Master's Thesis

Development of a Simulation Code for High-Power Cyclotrons

Chul Un Choi

Department of Physics

Graduate School of UNIST

2018

Development of a Simulation Code for High-Power Cyclotrons

Chul Un Choi

Department of Physics

Graduate School of UNIST

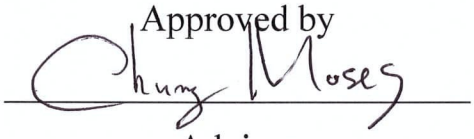
Development of a Simulation Code for High-Power Cyclotrons

A thesis
submitted to the Graduate School of UNIST
in partial fulfillment of the
requirements for the degree of
Master of Science

Chul Un Choi

1. 8. 2018 of submission

Approved by



Advisor

Moses Chung

Development of a Simulation Code for High-Power Cyclotrons

Chul Un Choi

This certifies that the thesis of Chul Un Choi is approved.

1/8/2018

signature

Advisor: Moses Chung

signature

typed name: min-sup Hur

signature

typed name: Seunghwan Shin

signature

Abstract

A high-power cyclotron is a good candidate as a machine of the ADS (Accelerator Driven System) for the transmutation of long lived nuclear wastes. A high-power cyclotron for ADS require about energy of 1 to 2GeV and 10 mA currents. To develop high power cyclotrons, simulation code for beam dynamics in cyclotron should also be developed by including higher order terms in transverse transfer matrix, accelerating effects and space charge effects. The simulation code can describe equivalent orbit at each energy by first order terms, but we have added acceleration effects and second order terms. Therefore, we expect to describe the beam motion more accurately. So, we first study fundamental beam dynamics in cyclotrons. Second, we explain equilibrium orbit code which is the present code in cyclotron. Third, the updated acceleration effects and second order terms in simulation code are introduced and results of the code for the simulation of a 13 MeV cyclotron are reported. Lastly, an upgrade plan is discussed.

Contents

List of Figures

List of Table

Chapter 1. Introduction: Why Cyclotron?

At present, the problem of disposal of nuclear waste comes to the fore in Korea, because it is difficult to secure site for storing nuclear waste. Nuclear waste which is a high-level radioactive material is the trash left after nuclear fission. So, it is sealed in a high-impact metal container, and then place in a nuclear waste disposal site constructed on a solid rock bed at a depth of several hundred meters or more below the ground to completely isolate it from the ecosystem. However, since this nuclear waste has a half-life of a few hundred thousand years, it should stay at the disposal site permanently. Especially in the case of Korea, the land where nuclear waste can be buried is limited. Therefore, there is a need to eliminate nuclear waste in addition to burying it underground. One of the ways to solve this problem is the ADS (Accelerator Driven System) [1].

The ADS consists of a high-power accelerator, a target for the nuclear destruction reaction and a subcritical fission reactor. When high energy protons hit on the target, high speed neutron is generated by the nuclear spallation, and the high-speed neutron is injected into the subcritical nuclear reactor. Finally, nuclear waste which is a high-level radioactive material can be converted into a stable nuclide by a fission reaction. Fission reaction also generates fission energy. When converting fission energy to electrical energy, it can be used as commercial power. Therefore, the ADS can be said that it functions as a kind of waste recycling. To maximize the generation of neutrons when the protons hit on the target, a high-power accelerator with an energy of 1 GeV and an output current of about 10 mA is required. Linear accelerators and cyclotrons are good candidates as a driver of the ADS. The advantages and disadvantages of the two accelerators are as follows. Linear accelerators are easy to manufacture and easy to extract beam. However, large sites are needed, and production costs are high. On the other hand, cyclotrons are difficult to extract beam, but its size is small and production costs are low. Therefore, cyclotrons are considered as proper accelerators for proper ADS in Korea, considering the shortage of land.

The cyclotrons are a type of circular accelerator. The first cyclotron concept was made by E. O. Lawrence. The first cyclotron has uniform magnets and two Dees which supply RF power (Fig.1.1). Because this cyclotron does not focus the beam, it outputs a beam of limited energy. There are three problems in outputting a beam of high energy. First, a uniform magnetic field is required to satisfy isochronism. Second, decreasing magnetic field with radius is required to focus beam. Third, increasing magnetic field is needed to compensate relativistic effect. To solve these problems, sector focused cyclotron was invented by L. H. Thomas. This cyclotron uses an azimuthal varying magnetic field, and can retained the orbit stability while maintaining the isochronism. And to satisfy the isochronism, it increases the magnetic field according to the radius even at high energy. The sector focused cyclotron is also called the azimuthal varying field (AVF) cyclotron, and the energy limit is

70 ~ 100 MeV. More axial focusing is needed to overcome the energy limitation. The spiral sector cyclotron for more axial focusing force was developed in 1974 at the TRIUMF [2]. This cyclotron outputs the 500 MeV of energy. To maximize the focusing force, separated sector cyclotron was developed in 1974 at the Paul Scherrer Institute (PSI) [3]. In this cyclotron, the field of the valley part is 0, and the magnetic field exists only in the hill part. This cyclotron also outputs about 590 MeV of energy. Also, the superconducting cyclotrons have been developed.

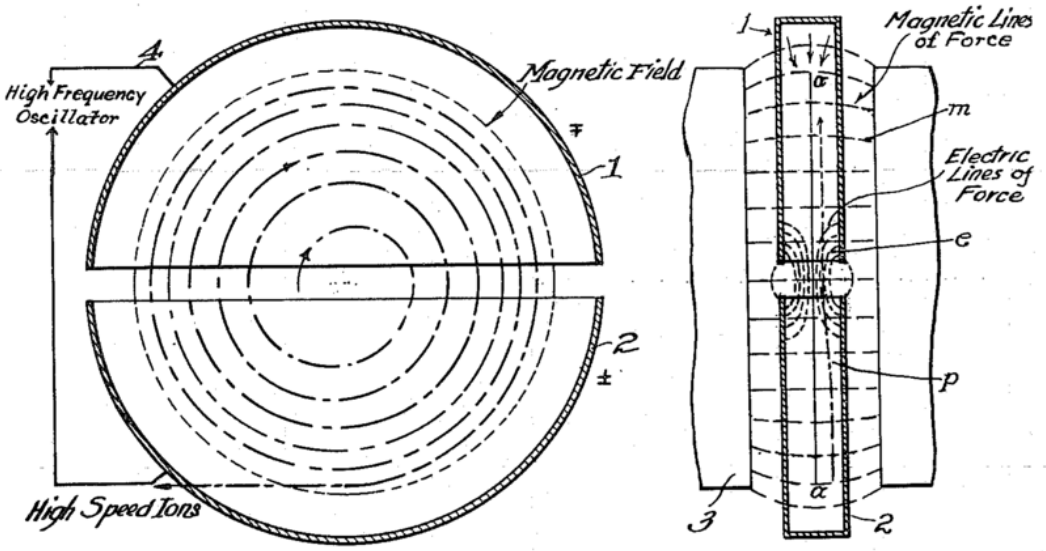


Fig. 1.1. The first cyclotron was invented by E. O. Lawrence

To make a high-power cyclotron, various research and development are required. In particular, the upgrading of the simulation code plays an essential role in calculating more accurate beam dynamics in the high-power cyclotron. Starting from the CYCLONE developed at Michigan state university in the 1960s, several cyclotron simulation codes have been developed and commercialized by Switzerland, the United States and other countries. In the past, equilibrium orbit codes such as CYCLOPS and GENSPEO were mainly developed due to limitations of computer performance. Now, as computer performance improves, codes with space charge effects and higher order terms have been developed.

In this paper, we improve the existing E.O code (Equilibrium Orbit code) and develop the beam tracking code by adding an acceleration effect and second order terms.

Chapter 2. Theory of Cyclotron

2.1 Basic of cyclotron

A cyclotron is a type of circular accelerator that accelerates charged particles into a spiral using high frequency electrodes and magnetic fields. Magnetic field is applied perpendicularly to the direction of the particle's motion, and causes the charged particle's path to bend in a circle due to Lorentz force. The charged particles are accelerated by using a high frequency alternating voltage which is applied between two hollow D-shaped electrodes called "Dees" inside a vacuum chamber.

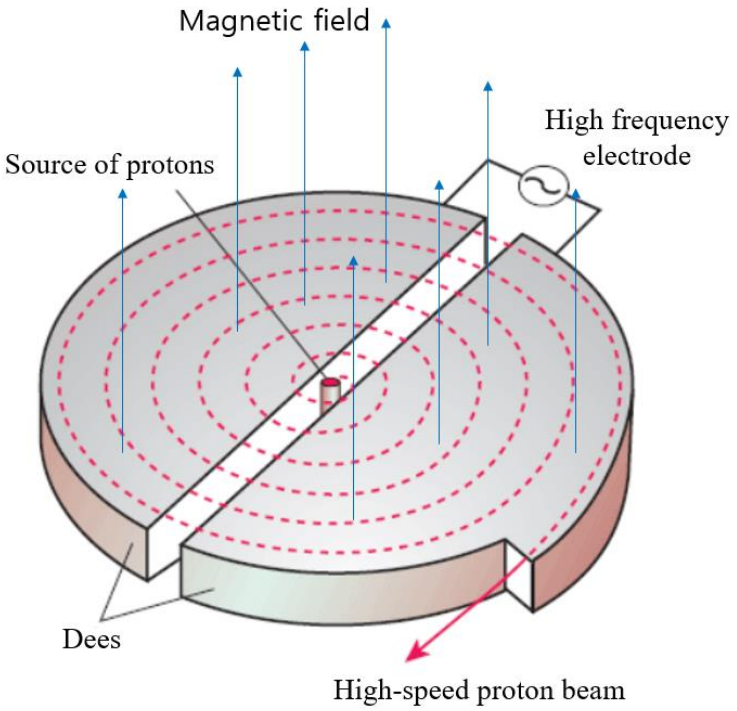


Fig. 2.1. Dees of cyclotron and particle's motion in cyclotron

The equations of particle's motion in magnetic field and electric field are given by

$$qvB = \frac{mv^2}{r}, \tag{2.1}$$

$$E = qV = \frac{1}{2}mv^2. \tag{2.2}$$

Using Eq. (1.1), angular velocity of particles is written as

$$\omega = \frac{v}{r} = \frac{qB}{m}. \tag{2.3}$$

To extract stable particles, magnet design is important in cyclotron. For the charged particles to have the maximum energy in dee gap, the isochronous condition in which RF frequency of dee and the angular frequency of particles are matched must be satisfied. The accelerated charged particles are so fast that relativistic effects should be considered. Therefore, as the velocity of particles increases, mass of particles also increases with radius. Mass of particle is given by

$$m(r) = m_o\gamma(r). \tag{2.4}$$

Here, m_o is rest mass of particle, γ is relativistic factor. Since the particle's mass increases along the radius, the angular velocity of particle becomes smaller in Eq. (2.3). If the angular velocity of particle becomes smaller, the isochronous condition is not satisfied. Therefore, a magnet having a larger magnetic field should be designed as the radius increases. But such a magnet defocuses the beam particles and they are quickly lost. To solve this defocusing effect, Thomas proposed introduction of sectors of high and low magnetic field in 1930s [4]. The magnetic field changes azimuthally by putting sectors. Because of this magnetic field, the beam is focused, and a high energy beam can be extracted. Thus, a cyclotron with such a magnetic field is sector focused cyclotron. In this study, we use the magnetic field data of sector focused cyclotron (Fig. 2.2).

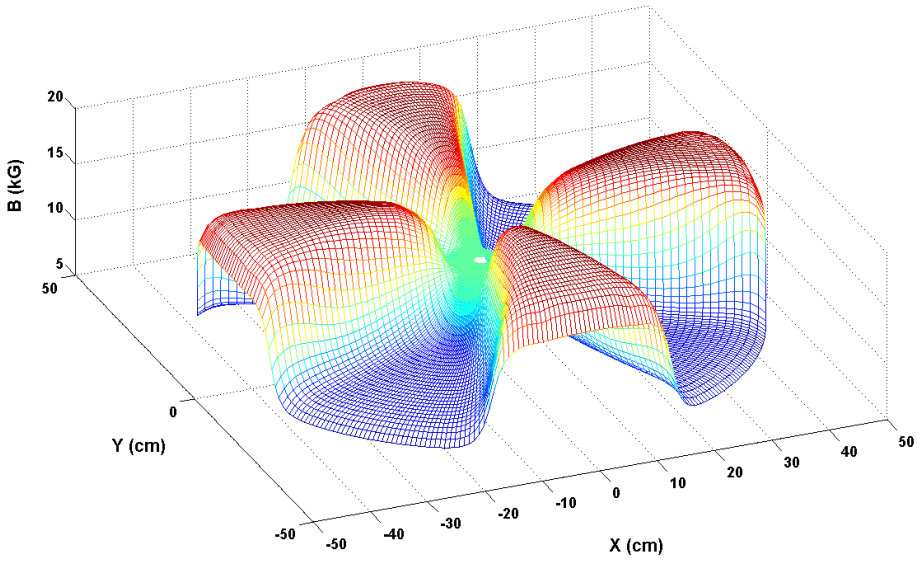


Fig. 2.2. Magnetic field of sector focused cyclotron (KIRAMS-13) [5].

2.2 Beam trajectory and beam focusing in sector focused cyclotron

The beam particle has a circular motion in a constant magnetic field. The orbit of this circular motion is called equilibrium orbit or closed orbit. Whether or not the magnet of cyclotron is well designed can be determined by closed orbit. Azimuthally varying magnetic field in sector focused cyclotron is written as

$$B(r,\theta) = B_o(r)(1 + f\cos N\theta). \tag{2.5}$$

Here, $B_o(r)$ is average magnetic field, f is flutter term that is rate of varying magnetic field for azimuth and *N* is number of sector. This magnet produces a non-circular orbit. Equation of motion in magnet is written as

$$\mathrm{m}\frac{d^2r}{dt^2} = qvB_o(1 + f\cos N\theta). \tag{2.6}$$

Radius of particles is proportional to $f\cos N\theta$ and is maximum in hill, which is the largest magnetic field region.

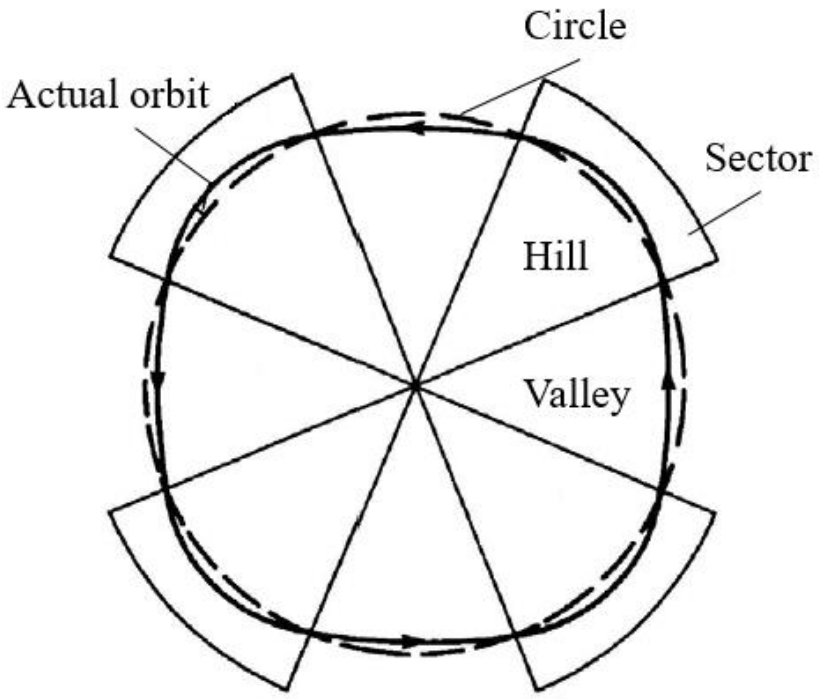


Fig. 2.3. Actual orbit of beam in four sector magnets

In the cyclotron, there is a restriction of the height of the vacuum chamber, so vertical focusing is important. At energy above a few MeV, the focusing by the electric field is not strong, so beam is focused by magnetic field except the central region of the cyclotron. In sector focused cyclotron, the beam is focused vertically by the influence of the sector. This phenomenon is called edge focusing.

Fig. 2.4. Top view of sector, s is Actual beam path, **σ** is vertical coordinate to sector and **α** is the angle of s and **σ**, Magnetic field is the direction from the paper and s is actual beam path.

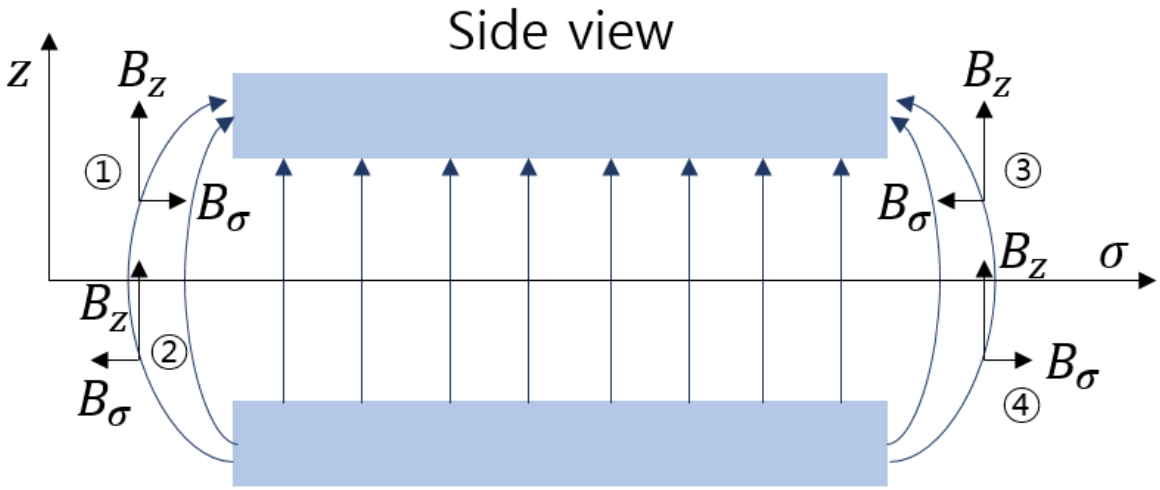


Fig. 2.5. Side view of sector. The beam goes from left to right and edge of the sector magnetic field is curved.

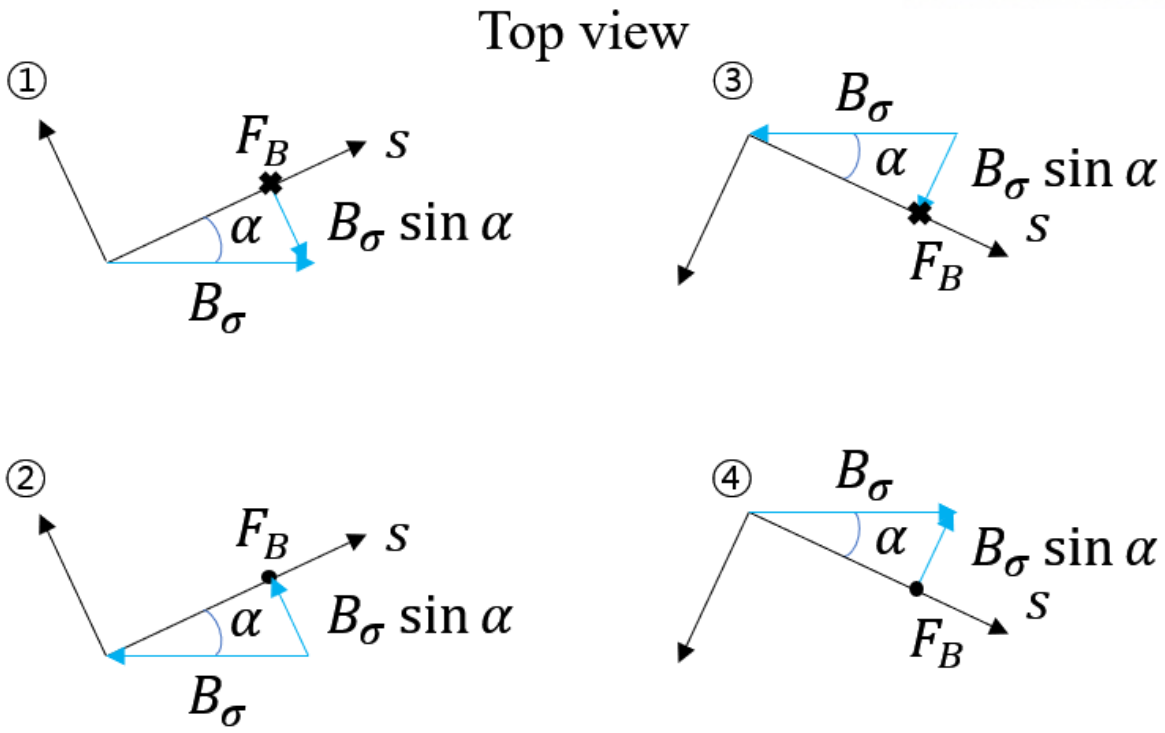


Fig. 2.6. From top view of the sector, it shows the direction of the beam and the Lorentz force. ①, ②, ③ and ④ are position in ①, ②, ③ and ④ of Fig. 2.5. The dot means that the direction of Lorentz force has component from the paper in ② and ④, the cross that the direction of Lorentz force has component towards the paper.

At the end of the sector, the magnetic field bends as shown in Fig. 2.5. The curved magnetic field causes edge focusing. As shown in Fig. 2.4, the beam is focused by the Lorentz force when it is incident obliquely to the sector, and when beam is incident vertically ($\alpha = 0$) to the sector, it is not focused. In Fig. 2.6, at position ① and ③, the beam is forced downward by the Lorentz force (toward the center of the beam), while the beam at position ② and ④ is force upward (toward the center of the beam). The Lorentz force on the beam is written as

$$F_B = qvB_\sigma \sin\alpha. \tag{2.7}$$

Therefore, the beam entering the sector and exiting the sector is focused. When the beam repeats the process passing through the sector, the beam is focused by the calculation of the matrix equation.

2.3 Matrix equation in hill and valley

The equation of motion for charged particles around equilibrium orbit in magnetic field [6] is given by:

$$x''(s) + \kappa(s)x(s) = 0, \tag{2.8}$$

where primes indicate derivatives with respect to the axial distance 's' measured along the equilibrium orbit, $s = \rho\theta$, where ρ is radius. $\kappa(s)$ is focusing function and is related to the magnetic field. the $\kappa(s)$ is represented by k for vertical motion and $1-k$ for horizontal motion. The range of k is given by $0 < k < 1$. Equation (2.8), known as Hill's equation, is a homogeneous, second order ordinary differential equation with a coefficient, the focusing function, which is not constant in general.

For now, we consider horizontal component of the transverse coordinates for particle motion. The solution depends on the initial conditions $x(0)$, $x'(0)$ and $\kappa(s)$. The general solution of Eq. (2.8) can be written as a superposition of cosine-like and sine-like function [7]:

$$x = A\cos\sqrt{1-k}\theta + B\sin\sqrt{1-k}\theta, \tag{2.9}$$

$$x' = \sqrt{1-k}(-A\sin\sqrt{1-k}\theta + B\cos\sqrt{1-k}\theta), \tag{2.10}$$

which for $x(0) = x_0$ and $x'(0) = x_0' = \frac{1}{\rho}\frac{dx}{d\theta}_{\theta=0}$, become

$$x = (\cos\sqrt{1-k}\theta)x_0 + \frac{\rho}{\sqrt{1-k}}(\sin\sqrt{1-k}\theta)x_0', \tag{2.11}$$

$$x' = -\frac{\sqrt{1-k}}{\rho}\left(\sin\sqrt{1-k}\theta\right)x_0 + (\cos\sqrt{1-k}\theta)x_0'. \tag{2.12}$$

Equations (2.11) and (2.12) can be represented by matrix:

$$\begin{pmatrix} x \\ x' \end{pmatrix} = \begin{pmatrix} \cos\sqrt{1-k}\theta & \frac{\rho}{\sqrt{1-k}}(\sin\sqrt{1-k}\theta) \\ -\frac{\sqrt{1-k}}{\rho}\left(\sin\sqrt{1-k}\theta\right) & \cos\sqrt{1-k}\theta \end{pmatrix}\begin{pmatrix} x_0 \\ x_0' \end{pmatrix} = M_H(\theta)\begin{pmatrix} x_0 \\ x_0' \end{pmatrix}. \tag{2.13}$$

Similarly, matrix in the vertical component is written as

$$\begin{pmatrix} z \\ z' \end{pmatrix} = \begin{pmatrix} cos\sqrt{k}\theta & \frac{\rho}{\sqrt{k}}(sin\sqrt{k}\theta) \\ -\frac{\sqrt{k}}{\rho}(sin\sqrt{k}\theta) & cos\sqrt{k}\theta \end{pmatrix} \begin{pmatrix} z_0 \\ z_0' \end{pmatrix} = M_V(\theta) \begin{pmatrix} z_0 \\ z_0' \end{pmatrix}. \quad (2.14)$$

When the beam passes through the boundary between hill and valley, the magnetic field at the edge of the sector acts as a thin lens and can be approximated as $\theta \to 0$. Therefore, the horizontal component of matrix M_H is written as

$$M_{H\theta\to 0} = \begin{pmatrix} 1 & 0 \\ -\sqrt{1-k}\theta & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ -\frac{1}{f} & 1 \end{pmatrix}. \quad (2.15)$$

f is focal length and sign of focal length is determined by the angle between the vertical direction of the sector boundary and the beam path.

One sector consists of a boundary, hill, boundary and valley. The matrix of vertical component when the beam passes through on sector is as follows:

$$M_{hill} = \begin{pmatrix} 1 & 0 \\ \frac{1}{f} & 1 \end{pmatrix} \begin{pmatrix} cos\sqrt{k}\theta & \frac{\rho}{\sqrt{k}}(sin\sqrt{k}\theta) \\ -\frac{\sqrt{k}}{\rho}(sin\sqrt{k}\theta) & cos\sqrt{k}\theta \end{pmatrix} \begin{pmatrix} 1 & 0 \\ \frac{1}{f} & 1 \end{pmatrix}, \quad (2.16)$$

$$M_{valley} = \begin{pmatrix} 1 & 0 \\ -\frac{1}{f} & 1 \end{pmatrix} \begin{pmatrix} cos\sqrt{k}\theta & \frac{\rho}{\sqrt{k}}(sin\sqrt{k}\theta) \\ -\frac{\sqrt{k}}{\rho}(sin\sqrt{k}\theta) & cos\sqrt{k}\theta \end{pmatrix} \begin{pmatrix} 1 & 0 \\ -\frac{1}{f} & 1 \end{pmatrix}, \quad (2.17)$$

$$M = M_{hill}M_{valley}, \quad (2.18)$$

$$\begin{pmatrix} y \\ y' \end{pmatrix} = M_{hill}M_{valley} \begin{pmatrix} y_0 \\ y_0' \end{pmatrix} = M \begin{pmatrix} y_0 \\ y_0' \end{pmatrix}. \quad (2.19)$$

Therefore, when the particle is rotated one turn in cyclotron consisting of N sector, the vector of the particle is as follows:

$$Y_0 = \begin{pmatrix} y_0 \\ y_0' \end{pmatrix}, \quad (2.20)$$

$$Y_N = M^N Y_0. \quad (2.21)$$

The component of matrix at each point are different due to the change in magnetic field along the theta and radius. But, note that the Liouville's theorem leads to the value for the determinant

$$|M| = M_{11}(\theta)M_{22}(\theta) - M_{12}(\theta)M_{21}(\theta) = 1. \quad (2.22)$$

2.4 Focusing frequency (tune) and resonance

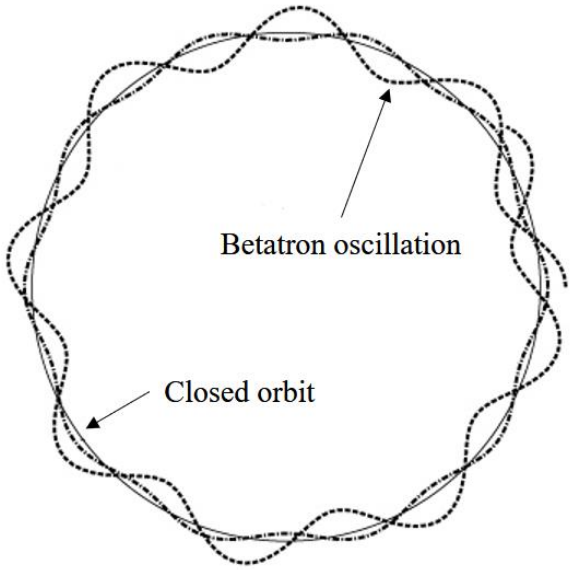


Fig. 2.7. Betatron oscillation. The tune is the number of cycles of the oscillation in one complete orbit around ring.

The particles around the beam, which are an ensemble of particles, also oscillate around the equilibrium orbit. This oscillation is called betatron oscillation, which we see as a sinusoidal behavior in Fig. 2.7. Also, the betatron tune is defined as the number of cycles of the betatron oscillation made by a particle during on full orbit. We can calculate the betatron tunes with Eqs. (2.13) and (2.14),

$$cos\sqrt{1-k}\theta = \frac{1}{2}(M_{11}+M_{22}) = \frac{1}{2}Tr(M_H), \tag{2.23}$$

$$cos\sqrt{k}\theta = \frac{1}{2}(M_{11}+M_{22}) = \frac{1}{2}Tr(M_v). \tag{2.24}$$

The horizontal and vertical betatron tunes are written as

$$Q_H = \frac{\sqrt{1-k}2\pi}{2\pi} = \sqrt{1-k},\ Q_v = \frac{\sqrt{k}2\pi}{2\pi} = \sqrt{k}. \tag{2.25}$$

The amplitude of betatron oscillations grows at a resonance. When resonances occur, the beam becomes unstable, causing a lot of beam loss and damaging the cyclotron. The resonances occur when the errors are encountered at the same phase of betatron oscillation during each revolution and the coupling between horizontal and vertical motion.

The resonance condition [8] is written as

$$KQ_r \pm LQ_z = pN, \tag{2.26}$$

where K, L, p is integers and N is number of sector. $|K|+|L|$ is called the resonance order. To operate the beam stably, it is necessary to avoid the resonance that satisfies the above conditions. In particular, coupling $Q_r = 2Q_z$, called, the Walkinshaw resonance contributes greatly to the instability of the beam. To reduce this resonance effect, the amplitude of the beam in the horizontal direction should be made small.

2.5 RF System

In the cyclotrons, RF power is applied at the dee gap to accelerate the particles. At this time, the phase of the RF frequency and the phase of the angular frequency of the particles should match, so that the particles are accelerated by receiving the maximum voltage [9]. If the phases are not matched, a phase shift occurs, and the particles may not accelerate properly or may decelerate. The angular frequency of the particles is proportional to the magnetic field at

$$\omega_{rev} = \frac{qB}{m_0\gamma(r)}, \tag{2.27}$$

so the design of the magnet affects it.

In the real cyclotrons, since the beam bunch (which is not a single particle) passes through the Dee gap, if the preceding particles receive the maximum voltage and passes the Dee gap, the particles located behind are accelerated to a voltage smaller than preceding particles. Therefore, the beam bunch will be longer, and the number of decelerated particles will increase. So, when passing through the Dee gap, the front particles in the beam bunch should pass through the accelerating phase with a voltage less than the particles located behind in the beam bunch. However, this thesis does not deal with phase shift.

Chapter 3. Equilibrium Orbit Code

Given the magnetic field $B(r, \theta)$ in polar coordinates, the E.O code (Equilibrium Orbit code) calculates at each energy with optimum efficiency all the important properties of the closed orbits and of the linear horizontal and vertical oscillations about those orbits. This code is based on direct numerical integration of first order equations of motion of particles. Also, this code includes an iterative scheme, which reduces error of numerical integration.

The algorithm of E.O code is shown in Fig. 3.1.

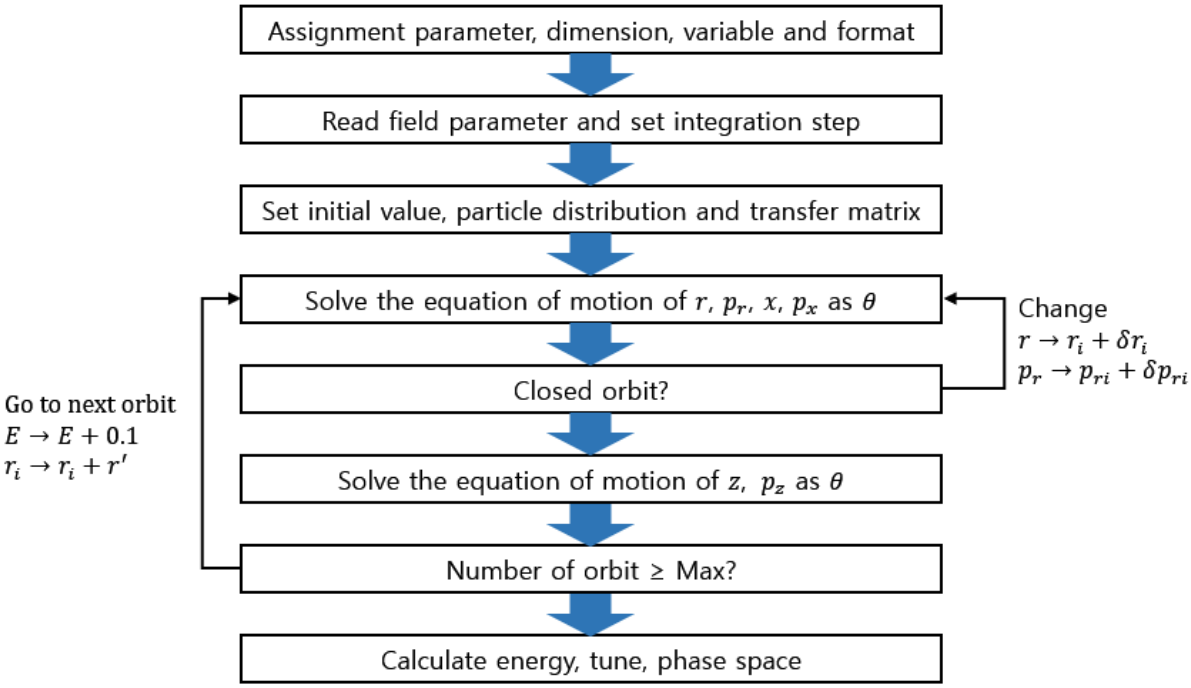


Fig. 3.1. Algorithm of E.O code

For computational efficiency, the code calculates the horizontal direction of equation of motion first. The code adjusts the initial conditions of r and p_r until the closed orbit condition is satisfied, and then it calculates the equation of motion in the vertical direction. After finishing the calculation for one orbit, it goes to the next energy orbit and repeats the above process. Once calculating up to the final orbit, the code can compute the energy, tune and phase space at each orbit.

3.1 Derivation of the canonical equation of motion in cyclotron

When the particle with charge q moves in the magnetic field with vector potential $\vec{A}$, the canonical momentum of that particle $\vec{P}$ is given by:

$$\vec{P} = \vec{p} + q\vec{A}. \tag{3.1}$$

Then, the action integral in a cylindrical coordinate is expressed as

$$\begin{aligned} \mathrm{S} &= \int_{t_o}^{t} \mathcal{L} dt = \int_{t_0}^{t} \left(P_x \frac{dx}{dt} + P_y \frac{dy}{dt} + P_z \frac{dz}{dt} - \mathcal{H} \right) dt \\ &= \int_{\theta_0}^{\theta} \left(P_r \frac{dr}{d\theta} + P_\theta \frac{d\theta}{d\theta} + P_z \frac{dz}{dt} - \mathcal{H} \frac{dt}{d\theta} \right) d\theta = \int_{\theta_0}^{\theta} \left(P_r \frac{dr}{d\theta} + P_z \frac{dz}{dt} - \mathcal{H} \frac{dt}{d\theta} + P_\theta \right) d\theta. \end{aligned} \tag{3.2}$$

Provided that particle coordinate increases in time, we can change the role of that coordinate and time. Therefore, form Eq. (3.2), if we choose angle θ instead of time t as independent variable [10] canonical momenta are written as

$$\begin{aligned} P_r &= p_r + qA_r, \\ P_\theta &= rp_\theta + qrA_\theta, \\ P_z &+ p_z + qA_z, \end{aligned} \tag{3.3}$$

and the new Hamiltonian is written as

$$\mathcal{H} = -P_\theta = -rp_\theta - qrA_\theta = -r\sqrt{(P^2 - (P_r - qA_r)^2 - (P_z - qA_z)^2)} - qrA_\theta. \tag{3.4}$$

The E.O code uses a magnetic field with a value divided by charge q $(\vec{B} \to \vec{B}/q)$, then the equation of motion can be obtained by calculating Hamiltonian equation with new Hamiltonian of Eq. (3.4):

$$\frac{dr}{d\theta} = \frac{\partial \mathcal{H}}{\partial P_r} = \frac{r(P_r - A_r)}{\sqrt{(P^2 - (p_r - qA_r)^2 - (P_z - qA_z)^2)}} = \frac{rp_r}{\sqrt{(P^2 - p_r^2 - p_z^2)}} = \frac{rp_r}{p_\theta}, \tag{3.5}$$

$$\frac{dp_r}{d\theta} = -\frac{\partial \mathcal{H}}{\partial r} - \frac{\partial A_r}{\partial r}\frac{dr}{d\theta} - \frac{\partial A_r}{\partial z}\frac{dz}{d\theta} - \frac{\partial A_r}{\partial \theta} = p_\theta - r\left(B - \frac{1}{2}z^2\left(\vec{\nabla}_\perp^2 B\right)\right) + \frac{\mathrm{zp_z}}{p_\theta}\frac{\partial B}{\partial \theta}, \tag{3.6}$$

$$\frac{dz}{d\theta} = \frac{\partial \mathcal{H}}{\partial P_z} = \frac{r(P_z - A_z)}{\sqrt{(P^2 - (p_r - qA_r)^2 - (P_z - qA_z)^2)}} = \frac{rp_z}{\sqrt{(P^2 - p_r^2 - p_z^2)}} = \frac{rp_z}{p_\theta}, \tag{3.7}$$

$$\frac{dp_z}{d\theta} = -\frac{\partial \mathcal{H}}{\partial z} - \frac{\partial A_z}{\partial r}\frac{dr}{d\theta} - \frac{\partial A_z}{\partial z}\frac{dz}{d\theta} - \frac{\partial A_z}{\partial \theta} = z\left(r\frac{\partial B}{\partial r} - \frac{p_r}{p_\theta}\frac{\partial B}{\partial \theta}\right). \tag{3.8}$$

From Eqs. (3.5) and (3.6), equation of EO at $z = 0$ are expressed as

$$\frac{dr}{d\theta} = \frac{rp_r}{p_\theta}, \tag{3.9}$$

$$\frac{dp_r}{d\theta} = p_\theta - rB. \tag{3.10}$$

We expand coordinates such that $r \to r + x,\ p_r \to p_r + p_x$ where $r,\ p_r$ are values in the equilibrium orbit. By expanding equations of displaced orbit in terms of r and p_r, the horizontal components of beam equations (see Appendix 7.1) are given by:

$$r_1' = \frac{p_r}{p_\theta}x + \frac{rP^2}{p_\theta^3}p_x, \tag{3.11}$$

$$p_{r1}' = -\frac{p_r}{p_\theta}p_x - \left(B + r\frac{\partial B}{\partial r}\right)x. \tag{3.12}$$

Here, r_1' and p_{r1}' are $\frac{dx}{d\theta}$ and $\frac{dp_x}{d\theta}$ which are horizontal components of beam coordinates. The E.O code expand the differential equations to first order terms only.

3.2 Iteration process

We require a map of the median-plane field of cyclotron. We input the field data $\mathrm{B}(\mathrm{r}, \theta)$ on a polar mesh.

The equilibrium orbit code must find the closed orbit by finding solution of the equation of motion at each energy. The closed orbit is the normal equilibrium orbit in a field having N sectors and no imperfections, also closed orbit is determined when it satisfies periodicity of

$$r(\theta + \theta_0) = r(\theta),\ p_r(\theta + \theta_0) = p_r(\theta). \tag{3.13}$$

The E.O code calculates differential equations of radius and radial momentum, Eqs. (3.9) and (3.10), as a Runge-Kutta Gill method which is a numerical integral in the field. In the E.O code, the integration interval for these differential equations is $\Delta\theta = 2°$. Continue to integrate the Eqs. (3.9) and (3.10) and become equal to the same value as the initial value at the end of integration. The conditions satisfy

$$r(\theta_i + \theta_0) = r_i,\ p_r(\theta_i + \theta_0) = p_{ri}, \tag{3.14}$$

As above, we also integrate the horizontal component of beam coordinate, Eqs. (3.11) and (3.12). We track the horizontal component of beam coordinate by using transfer matrix. To generate the transfer matrix, we set two independent solutions of these equations, denoted by (x_1, p_{x_1}) and (x_2, p_{x_2}). Initial conditions of two independent solutions are written as

$$\begin{aligned} x_1(\theta_i) &= 1,\ p_{x_1}(\theta_i) = 0, \\ x_2(\theta_i) &= 0,\ p_{x_2}(\theta_i) = 1, \end{aligned} \tag{3.15}$$

and the transfer matrix [11] from θ_i to a given θ is defined by

$$X(\theta, \theta_i) = \begin{pmatrix} x_1 & x_2 \\ p_{x_1} & p_{x_2} \end{pmatrix}_\theta. \tag{3.16}$$

Therefore, the general solution of Eqs. (3.11) and (3.12) can be written in matrix form as

$$\begin{pmatrix} x \\ p_x \end{pmatrix}_\theta = X(\theta, \theta_i) \begin{pmatrix} x \\ p_x \end{pmatrix}_i. \tag{3.17}$$

Also, transfer matrix $X(\theta, \theta_i)$ satisfies Liouville's theorem where

$$\|X(\theta, \theta_i)\| = x_1(\theta) p_{x_2}(\theta) - x_2(\theta) p_{x_1}(\theta) \equiv 1. \tag{3.18}$$

In Eqs. (3.11) and (3.12), $(x, p_x)_i$ are the horizontal position and horizontal direction momentum of the initial particle, and we can set the number and distribution of particles as desired in Fig. 3.2. Also, $(x, p_x)_\theta$ can be obtained by multiplying the transfer matrix by $(x, p_x)_i$.

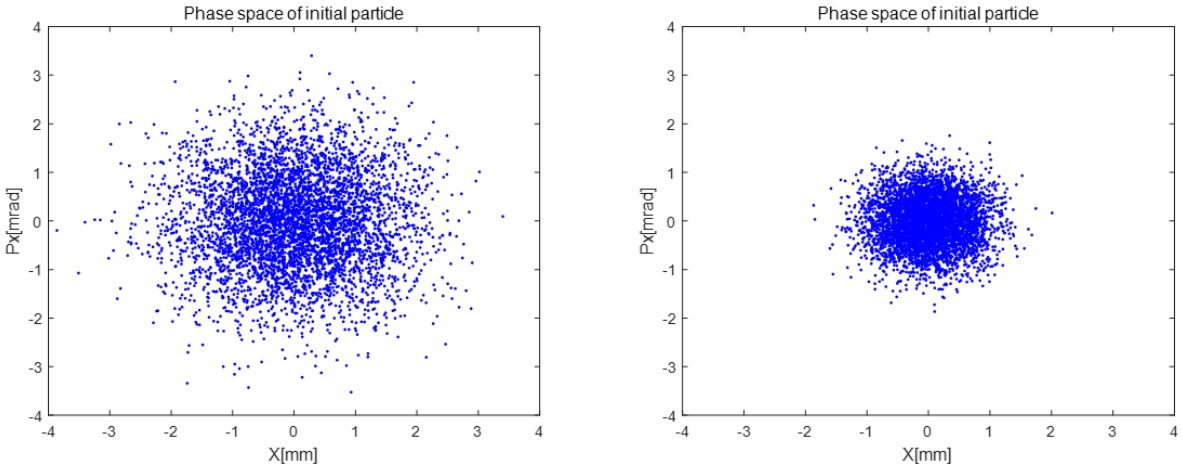


Fig. 3.2. Phase space of initial particles. The standard deviations of the left graph are $\boldsymbol{\sigma_x, = 1\ mm}$, $\boldsymbol{\sigma_{p_x} = 1\ mrad}$. The right graph of standard deviation is $\boldsymbol{\sigma_x = 0.5\ mm, \sigma_{p_x} = 0.5\ mrad}$. The number of particles is 5000 and the particles have Gaussian distributions.

Each cycle of the iteration process starts with a value (r_i, p_{r_i}). These initial conditions are used to carry out the integration of the differential equations noted above, namely, Eqs. (3.9) and (3.10) for r and p_r, the horizontal component of beam coordinates and Eqs. (3.11) and (3.12) for the elements of the matrix $X(\theta, \theta_i)$ of Eq. (3.16). All differential equations are integrated simultaneously.

When the integrations are completed at $\theta = \theta_f$, a numerical error occurs. This numerical error occurs when the start value and the end value do not match, and the error must be minimized. The errors in the trial values of (r_i, p_{r_i}) are calculated from

$$\epsilon_1 = r(\theta_f) - r_i,\ \epsilon_2 = p_r(\theta_f) - p_{ri}. \tag{3.19}$$

To reduce the error, we assume the closed orbit differ from the trial orbit. We can write $r_c(\theta_i) = r_i + \delta r_i$, and $p_{rc}(\theta_i) = p_{ri} + \delta p_{ri}$. (r_c, p_{rc}) and (r, p_r) are, respectively, the components of the closed orbit and the trial orbit, respectively, and δr_i and δp_{ri} are the correction values. Therefore, matrix can be written as

$$\begin{pmatrix} r_c \\ p_{rc} \end{pmatrix}_\theta = \begin{pmatrix} r \\ p_r \end{pmatrix}_\theta + \mathrm{X}(\theta, \theta_\mathrm{i}) \begin{pmatrix} \delta r_i \\ \delta p_{ri} \end{pmatrix}. \tag{3.20}$$

When $\theta = \theta_f$, Eq. (3.20) is written as

$$0 = \begin{pmatrix} \epsilon_1 \\ \epsilon_2 \end{pmatrix} + [\mathrm{X}(\theta, \theta_\mathrm{i}) - \mathrm{I}] \begin{pmatrix} \delta r_i \\ \delta p_{ri} \end{pmatrix}. \tag{3.21}$$

Then, we can calculate correction values which are

$$\begin{aligned} \delta r_i &= \frac{(X_{22}-1)\epsilon_1 - X_{12}\epsilon_2}{X_{11}+X_{22}-2}, \\ \delta p_{ri} &= \frac{(X_{11}-1)\epsilon_2 - X_{21}\epsilon_2}{X_{11}+X_{22}-2}. \end{aligned} \tag{3.22}$$

We can change the initial conditions

$$r_i \to r_i + \delta r_i,\ \delta p_{ri} \to p_{ri} + \delta p_{ri}. \tag{3.23}$$

Correction values are added to reduce the error. Improved values for the initial conditions are used to repeat the entire process in the next cycle, and the iteration stops when the error is smaller than ϵ_0 which is the pre-assigned error limit. Through this process, the correct initial values (r_i, p_{ri}) and closed orbit can be obtained.

Then, we integrate the differential equations in the vertical component of beam coordinate obtained by Eqs. (3.7) and (3.8). Those equations are simplified as

$$\begin{aligned} \frac{dz}{d\theta} &= \frac{rp_r}{p_\theta}, \\ \frac{dp_z}{d\theta} &= z\left(r\frac{\partial B}{\partial r} - \frac{p_r}{p_\theta}\frac{\partial B}{\partial \theta}\right). \end{aligned} \tag{3.24}$$

Similar to the horizontal component of beam coordinate, two independent solutions of these equations are required to generate the transfer matrix, denoted by (z_1, p_{z_1}) and (z_2, p_{z_2}). Initial conditions of two independent solutions are

$$\begin{aligned} z_1(\theta_i) = 1,\ p_{z_1}(\theta_i) = 0, \\ z_2(\theta_i) = 0,\ p_{z_2}(\theta_i) = 1, \end{aligned} \tag{3.25}$$

so that the transfer matrix $Z(\theta, \theta_i)$ from θ_i to a given θ is defined by

$$Z(\theta, \theta_i) = \begin{pmatrix} z_1 & z_2 \\ p_{z_1} & p_{z_2} \end{pmatrix}_\theta. \tag{3.26}$$

Also, the general solution of Eq. (3.26), can be written in matrix form as

$$\begin{pmatrix} z \\ p_z \end{pmatrix}_\theta = Z(\theta, \theta_i) \begin{pmatrix} z \\ p_z \end{pmatrix}_i. \tag{3.27}$$

The computation of matrix $Z(\theta, \theta_i)$ provides information about the linear vertical phase space and vertical oscillations, also the computation of matrix $X(\theta, \theta_i)$ provides information for the horizontal phase space and oscillations. When the integration finishes, the calculation for one orbit is completed, and we go to the next orbit by increasing energy and radius. Then the above iteration process is repeated.

3.3 Tune (Q_H and Q_v) calculation

The program can provide values of Q_H and Q_v which are number of linear oscillations for horizontal and vertical directions. The tune can be calculated using the transfer matrix we already defined [Eqs. (3.16) and (3.26)]. Also, in Sec. 2.4, we drive the tune using transfer matrix. We can set the transfer matrix using Eqs. (2.13) and (2.14) and Eqs. (3.17) and (3.27). Then the transfer matrix is written as

$$\begin{pmatrix} cos\,\sigma & \mathrm{A}\,sin\,\sigma \\ -\mathrm{A}\,sin\,\sigma & cos\,\sigma \end{pmatrix} = Y(\theta, \theta_i) = \begin{pmatrix} Y_{11} & Y_{12} \\ Y_{21} & Y_{22} \end{pmatrix}. \tag{3.28}$$

The transfer matrix $Y(\theta, \theta_i)$ stands for $X(\theta, \theta_i)$ or $Z(\theta, \theta_i)$, also in $\sigma = n\theta$, n stands for $\sqrt{k} = Q_H$ or $\sqrt{1-k} = Q_v$. At the end of the integration, $\theta_f = \theta_i + \theta_0$, we know that $Y(\theta_f, \theta_i)$ is the transfer matrix for on complete period starting at θ_i. Then we can calculate the betatron tune,

$$cos\,\sigma = \frac{1}{2}\left(Y_{11}(\theta_f, \theta_i) + Y_{22}(\theta_f, \theta_i)\right) = \frac{1}{2}Tr(Y), \tag{3.29}$$

by analogy to Eqs. (2.23) and (2.24). Also, betatron tune n is written as

$$n\frac{2\pi}{2\pi} = \cos^{-1}\left[\frac{1}{2}\left(Y_{11}(\theta_f, \theta_i) + Y_{22}(\theta_f, \theta_i)\right)\right]. \tag{3.30}$$

The program provides the output of tune at each orbit. We analyze the tune and check whether the magnetic field configuration is appropriate or not. When tune is an integer multiple, we should change the magnetic field data to avoid resonance. The resonance conditions are $KQ_r \pm LQ_z = pN$, Eq. (2.26), and the resonances causes beam loss and damage cyclotron. So, the transfer matrix components should be corrected so that the tune is not an integer, and the occurrence of resonance is suppressed. To correct these matrix components, it is necessary to modify the magnetic field data.

3.4 Result of simulation

We simulated the E.O code using 13 MeV cyclotron, KIRAMS-13's magnetic field data [12]. The KIRAMS-13 is a sector focused cyclotron and has four sectors. The Fig. 2.2 is KIRAMS-13's magnetic field data, and we input this data into E.O code. Also, we set initial energy $\mathrm{E} = 0.31$ MeV, $r_i = 2.6$ inch and $p_{r_i} = -0.03$ mrad, arbitrarily, and then the above iteration process (Sec. 3.2) will start. Whenever we go to the next orbit, the code gives $\Delta\mathrm{E} = 0.1$ MeV to particles, and when the particles reach around 13 MeV of energy, the code ends. At the end of the code we can see many linear output values in E.O code.

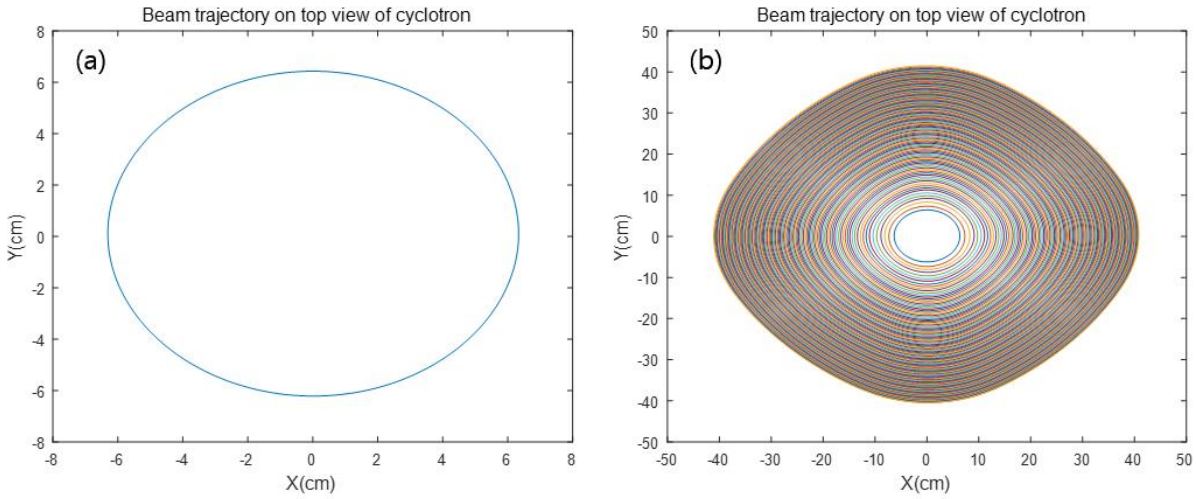


Fig. 3.3. (a) One closed orbit at 0.31MeV, (b) Closed orbits at 12.51 MeV.

Figure 3.3 shows the beam trajectory on top view of cyclotron, and from this figure we know the beam's trajectory according to each energy.

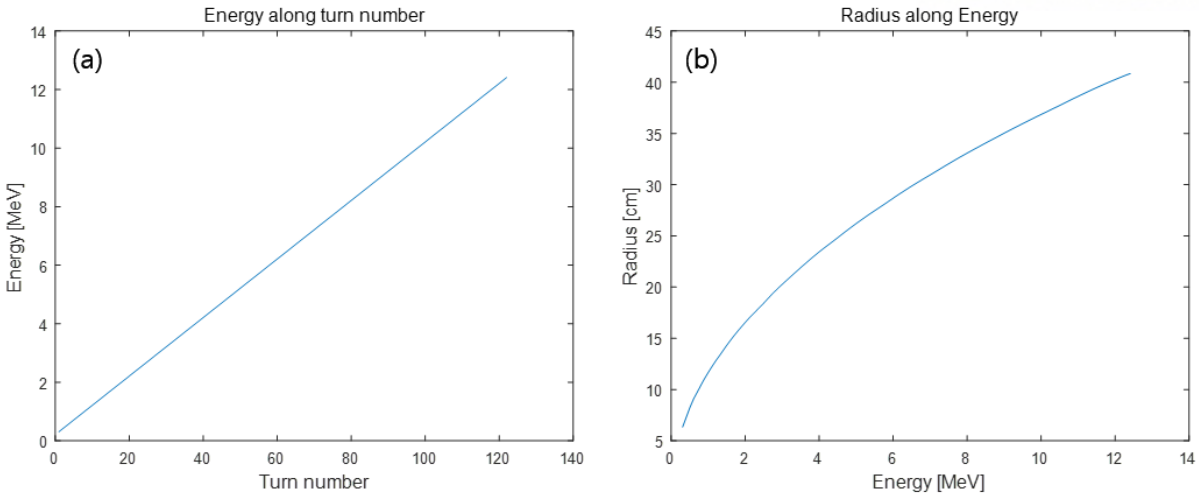


Fig. 3.4. (a) Energy along turn number, (b) Radius along the Energy.

In Fig. 3.4 (a), energy increases linearly. We simulate E.O code up to 122 turn numbers. The code adds 0.1 MeV of energy per turn. Figure 3.4 (b) is the graph of radius versus energy. It shows that radius is proportional to $\sqrt{E}$, which is consistent with the simple equations $\mathrm{E} = \frac{1}{2}mv^2$ and $qvB = \frac{mv^2}{r}$. Therefore, we can see that the final energy is 12.51 MeV and the final radius is 40.85 cm at 122 turns.

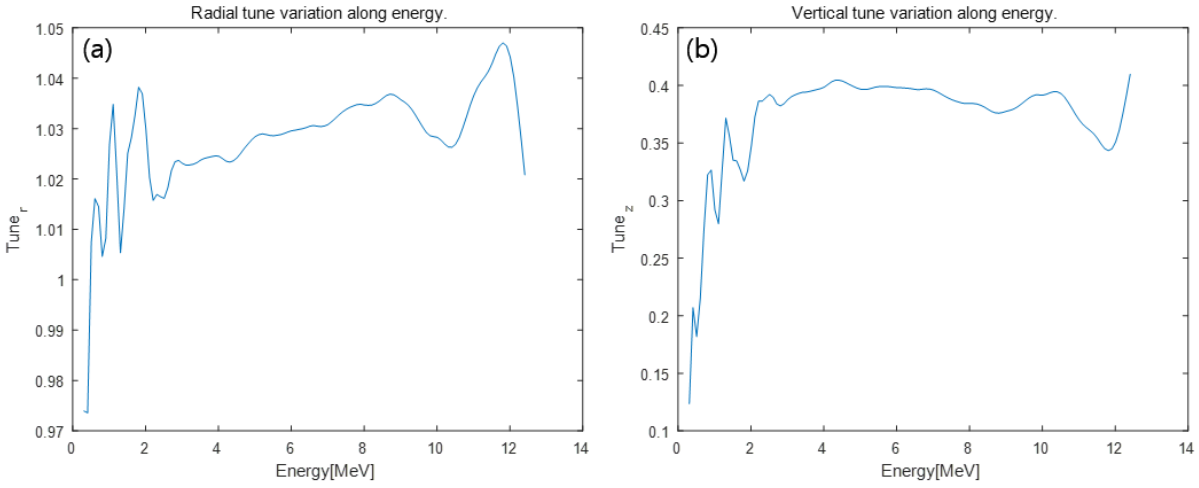


Fig. 3.5. (a) Radial tune variation along energy, (b) vertical tune variation along energy

The radial tune is approximately 0.98 to 1.05 and the vertical tune is approximately 0.4 to 0.5. The tune should avoid integers. Since the radial tune passes through quickly 1 at about 1 MeV, the beam can be operated stably in KIRAMS-13.

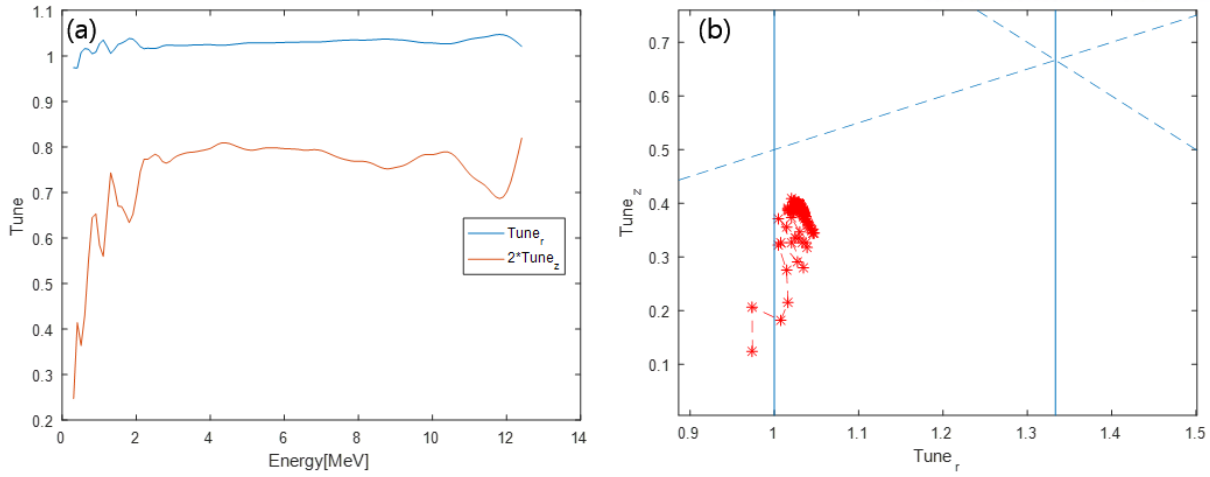


Fig. 3.6 (a) Tune distribution, avoiding dangerous resonance $\boldsymbol{Q_H = 2Q_v}$ (Walkinshaw resonance), (b) Tune diagram. The blue lines and dotted lines are the resonance lines to avoid, and the red dots are tunes of the beam at each energy.

We can see more clearly from Figs. 3.6(a) and 3.6(b) whether resonance occurs or not. The Fig. 3.6 (a) shows the Walkinshaw resonance which is a dangerous resonance. When the blue and orange lines overlap, a resonance occurs, and the beam will be lost. This figure shows that Walkinshaw resonance does not occur in KIRAMS-13.

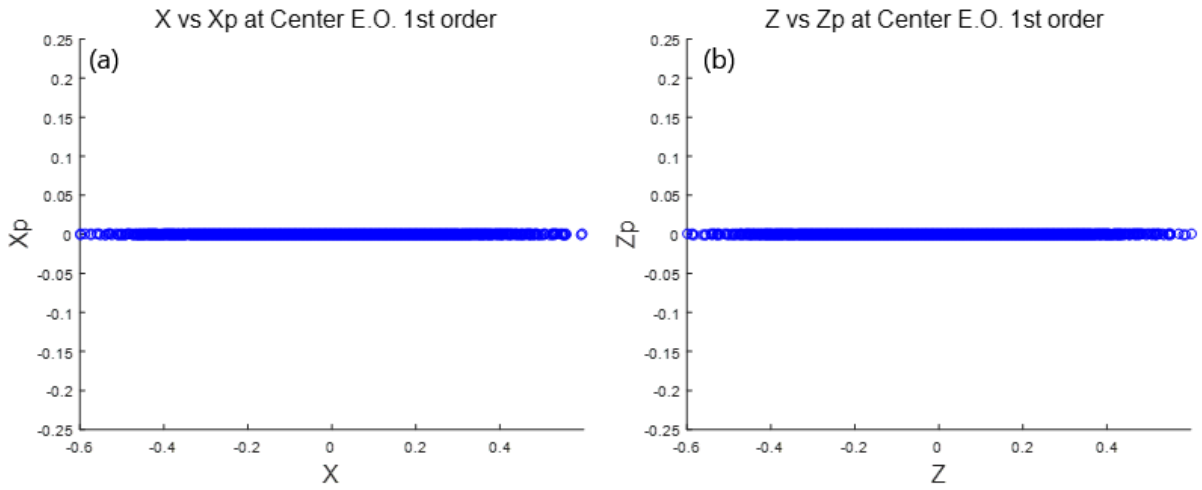


Fig. 3.7. Phase space of initial particles. In (a), graph of standard deviation is $\boldsymbol{\sigma_x = 0.05\ mm, \sigma_{p_x} = 0.0001\ mrad}$. In (b), graph of standard deviation is $\boldsymbol{\sigma_z = 0.05\ mm}$, $\boldsymbol{\sigma_{p_z} = 0.0001\ mrad}$. The number of particle is 5000 and particles have Gaussian distributions.

The Figs. 3.7(a) and 3.7(b) show initial conditions of particles and we represent the initial conditions as phase spaces of radial and vertical directions. The distribution of these particles is changed by the influence of the magnetic field as the beam moves in the cyclotron.

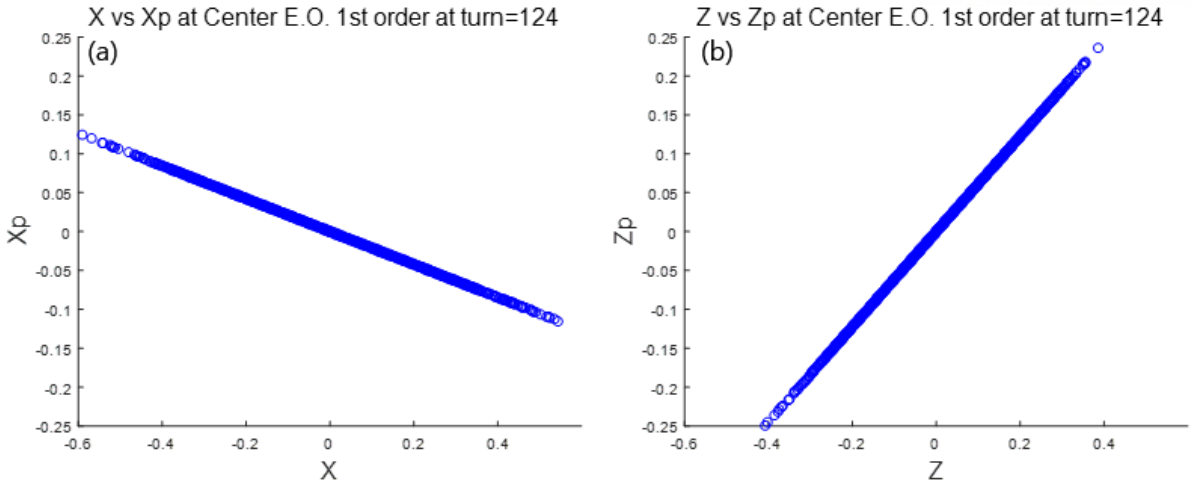


Fig. 3.8. Phase space of radial direction and vertical direction at 124 of turn number. The initial conditions are given in Figs. 3.7(a) and 3.7(b).

The Figs. 3.8(a) and 3.8(b) are phase space plots in radial and vertical directions when the energy is 12.71 MeV and the turn number is 124. By analyzing the phase space, we can figure out the stability of the beam and beam spread. As described earlier, the E.O code extends the beam equation of motion to the first order terms, so the beam performs a linear motion. When tracking the beam using E.O code, the phase space changes linearly, but the actual beam motion changes nonlinearly. In the E.O code, we can only confirm the linear motion of the beam by analyzing the phase space.

In Chapter 4, we will introduce the code that extends the beam equation of motion to the second order using Taylor expansion. In addition, we will compare the existing E.O code, second order code and full equation code, and finally explain characteristics of each code.

Chapter 4. Effects of Second Order Terms

We have updated the code by adding second order terms. This code is based on direct numerical integration of second order equations of motion of particles and extends the existing first order equations of motion to second order. Therefore, we can track the beam motion more accurately. Simulation code using full equations of motion takes a long time to compute beam evolution, because each particle is tracked by using full equations of motion, not using a transfer matrix technique. The second order code is more effective for particle tracking, because the computation time is faster than the code using full equations of motion, and beam tracking is more accurate than the E.O code.

In this chapter, we will study the second order equations of motion, describe the second order code, and compare it with E.O code and full equation code.

4.1 Derivation of the canonical second order equation of motion in cyclotron

In Sec. 3.1, we derived the equation of motion up to the first order. By expanding equations of motion in terms of x, p_x, z and p_z up to second order, second order equations of motion are derived as follows (see Appendix 7.2) [13]:

$$\frac{dx}{d\theta} = \frac{p_r}{p_\theta}x + \frac{rP^2}{p_\theta^3}p_x + \frac{\mathrm{P}^2}{p_\theta^3}xp_x + \frac{3}{2}\frac{rp_rP^2}{P_\theta^5}p_x^2 + \frac{1}{2}\frac{rp_r}{p_\theta^3}p_z^2, \tag{4.1}$$

$$\frac{dp_x}{d\theta} = -\frac{p_r}{p_\theta}p_x - \left(B + r\frac{\partial B}{\partial r}\right)x - \frac{1}{2}\frac{P^2}{p_\theta^3}p_x^2 - \frac{1}{2}\left(2\frac{\partial B}{\partial r} + r\frac{\partial^2 B}{\partial x^2}\right)x^2 - \frac{1}{2}\frac{p_z^2}{p_\theta} + \frac{1}{2}r\left(\vec{\nabla}^2 B\right)z^2 + \frac{\partial B}{\partial \theta}\frac{zp_z}{p_\theta}, \tag{4.2}$$

$$\frac{dz}{d\theta} = \frac{rp_z}{p_\theta} + \frac{1}{p_\theta}p_z x + \frac{rp_r}{p^3\theta}p_x p_z, \tag{4.3}$$

$$\frac{dp_z}{d\theta} = z\left(r\frac{\partial B}{\partial r} - \frac{p_r}{p_\theta}\frac{\partial B}{\partial \theta}\right) + \left(\frac{\partial B}{\partial r} + r\frac{\partial^2 B}{\partial r^2} - \frac{p_r}{p_\theta}\frac{\partial^2 B}{\partial r \partial \theta}\right)xz - \frac{p_r^2}{p_\theta^3}\frac{\partial B}{\partial \theta}zp_x. \tag{4.4}$$

Equations (4.1)-(4.4) are called non-linear equations of motion.

The transfer matrix can be easily formulated from the linear equation of motion in E.O code, but the matrix cannot be directly obtained from the second order equation of motion because the horizontal and vertical components are coupled. If we put transfer matrix

$$\begin{pmatrix} x \\ p_x \end{pmatrix}_\theta = \mathrm{X}(\theta, \theta_\mathrm{i})\begin{pmatrix} x \\ p_x \end{pmatrix}_i = \begin{pmatrix} X_{11}(\theta) & X_{12}(\theta) \\ X_{21}(\theta) & X_{22}(\theta) \end{pmatrix}\begin{pmatrix} x_0 \\ p_{x0} \end{pmatrix}, \tag{4.5}$$

$$\begin{pmatrix} z \\ p_z \end{pmatrix}_\theta = \mathrm{Z}(\theta, \theta_\mathrm{i})\begin{pmatrix} z \\ p_z \end{pmatrix}_i = \begin{pmatrix} Z_{11}(\theta) & Z_{12}(\theta) \\ Z_{21}(\theta) & Z_{22}(\theta) \end{pmatrix}\begin{pmatrix} z_0 \\ p_{z0} \end{pmatrix}. \tag{4.6}$$

to non-linear terms of Eqs. (4.1)-(4.4), we transform Eqs. (4.1)-(4.4) from non-linear equations to linear inhomogeneous equations (see Appendix 7.3), and obtain approximate solutions by solving the inhomogeneous equations. Inhomogeneous equation [Eq. (4.1)] is derived as below

$$\frac{dx}{d\theta} = \frac{p_r}{p_\theta}x + \frac{rP^2}{p_\theta^3}p_x + f_x(\theta),$$

$$\begin{aligned} f_x(\theta) &= \frac{\mathrm{P}^2}{p_\theta^3}(X_{11}x_0 + X_{12}p_{x0})(X_{21}x_0 + X_{22}p_{x0}) + \frac{3}{2}\frac{rp_rP^2}{P_\theta^5}(X_{21}x_0 + X_{22}p_{x0})^2 \\ &\quad + \frac{1}{2}\frac{rp_r}{p_\theta^3}(Z_{21}z_0 + Z_{22}p_{z0})^2 \\ &= \left(\frac{\mathrm{P}^2}{p_\theta^3}X_{11}X_{21} + \frac{3}{2}\frac{rp_rP^2}{p_\theta^5}X_{21}^2\right)x_0^2 + \left(\frac{\mathrm{P}^2}{p_\theta^3}(X_{11}X_{22} + X_{12}X_{21}) + 3\frac{rp_rP^2}{P_\theta^5}X_{21}X_{22}\right)x_0p_{x0} \\ &\quad + \left(\frac{\mathrm{P}^2}{p_\theta^3}X_{12}X_{22} + \frac{3}{2}\frac{rp_rP^2}{P_\theta^5}X_{22}^2\right)p_{x0}^2 + \left(\frac{1}{2}\frac{rp_r}{p_\theta^3}Z_{21}^2\right)z_0^2 + \left(\frac{rp_r}{p_\theta^3}Z_{21}Z_{22}\right)z_0p_{z0} + \left(\frac{1}{2}\frac{rp_r}{p_\theta^3}Z_{22}^2\right)p_{zo}^2. \end{aligned} \tag{4.7}$$

The above equation of motion can be written as

$$\frac{dx}{d\theta} = \frac{p_r}{p_\theta}x + \frac{rP^2}{p_\theta^3}p_x + f_{11}x_0^2 + f_{12}x_0p_{x0} + f_{13}p_{x0}^2 + f_{14}z_0^2 + f_{15}z_0p_{z0} + f_{16}p_{z0}^2. \tag{4.8}$$

The equations of motion of other components can also be written similar to the equation of motion for x [Eq. (4.6)]. Therefore, inhomogeneous equations of motion for p_x, z and p_z are written as

$$\frac{dp_x}{d\theta} = -\frac{p_r}{p_\theta}p_x - \left(B + r\frac{\partial B}{\partial r}\right)x + f_{21}x_0^2 + f_{22}x_0p_{x0} + f_{23}p_{x0}^2 + f_{24}z_0^2 + f_{25}z_0p_{z0} + f_{26}p_{z0}^2, \tag{4.9}$$

$$\frac{dz}{d\theta} = \frac{rp_z}{p_\theta} + g_{11}x_0z_0 + g_{12}x_0p_{z0} + g_{13}p_{x0}z_0 + g_{14}p_{x0}p_{z0}, \tag{4.10}$$

$$\frac{dp_z}{d\theta} = z\left(r\frac{\partial B}{\partial r} - \frac{p_r}{p_\theta}\frac{\partial B}{\partial \theta}\right) + g_{21}x_0z_0 + g_{22}x_0p_{z0} + g_{23}p_{x0}z_0 + g_{24}p_{x0}p_{z0}\,. \tag{4.11}$$

The horizontal components [Eqs. (4.8) and (4.9)] can be expressed as a matrix is equal to

$$\begin{aligned} \begin{pmatrix} \frac{dx}{d\theta} \\ \frac{dp_x}{d\theta} \end{pmatrix} &= \begin{pmatrix} \frac{p_r}{p_\theta} & \frac{rP^2}{p_\theta^3} \\ -\left(B + r\frac{\partial B}{\partial r}\right) & -\frac{p_r}{p_\theta} \end{pmatrix}\begin{pmatrix} x \\ p_x \end{pmatrix} + \begin{pmatrix} f_{11} \\ f_{21} \end{pmatrix}x_0^2 + \begin{pmatrix} f_{12} \\ f_{22} \end{pmatrix}x_0p_{x0} + \begin{pmatrix} f_{13} \\ f_{23} \end{pmatrix}p_{x0}^2 + \begin{pmatrix} f_{14} \\ f_{24} \end{pmatrix}z_0^2 \\ &\quad + \begin{pmatrix} f_{15} \\ f_{25} \end{pmatrix}z_0p_{z0} + \begin{pmatrix} f_{16} \\ f_{26} \end{pmatrix}p_{z0}^2. \end{aligned} \tag{4.12}$$

If we consider above equation including only x_0^2 terms in non-linear part, Eq. (4.12) becomes

$$\begin{pmatrix} \frac{dx}{d\theta} \\ \frac{dp_x}{d\theta} \end{pmatrix} = K(\theta)\begin{pmatrix} x \\ p_x \end{pmatrix} + \begin{pmatrix} f_{11} \\ f_{21} \end{pmatrix} x_0^2, \tag{4.13}$$

where $K(\theta)$ is the transfer matrix. The solution of Eq. (4.13) can be found using a Green function. The inhomogeneous solution can be written as

$$\begin{pmatrix} x(\theta) \\ p_x(\theta) \end{pmatrix} = \left(\int_0^\theta X(\theta)X^{-1}(\theta') \begin{pmatrix} f_{11}(\theta') \\ f_{21}(\theta') \end{pmatrix} d\theta' \right) x_0^2. \tag{4.14}$$

Therefore, the solution of the equation of motion for second order including only x_0^2 term is

$$\begin{pmatrix} x(\theta) \\ p_x(\theta) \end{pmatrix} = \begin{pmatrix} X_{11}(\theta) & X_{12}(\theta) \\ X_{21}(\theta) & X_{22}(\theta) \end{pmatrix} \begin{pmatrix} x_0 \\ p_{x0} \end{pmatrix} + \left(\int_0^\theta X(\theta)X^{-1}(\theta') \begin{pmatrix} f_{11}(\theta') \\ f_{21}(\theta') \end{pmatrix} d\theta' \right) x_0^2, \tag{4.15}$$

and the vertical components are also calculated as the above equation (see Appendix 7.3). To derive the complete solution of the equation of motion for second order, we define A_n and B_n as

$$\begin{aligned} A_n &= \begin{pmatrix} a_{1n} \\ a_{2n} \end{pmatrix} = \int_0^\theta X(\theta)X^{-1}(\theta') \begin{pmatrix} f_{1n}(\theta') \\ f_{2n}(\theta') \end{pmatrix} d\theta', \\ b_n &= \begin{pmatrix} b_{1n} \\ b_{2n} \end{pmatrix} = \int_0^\theta Z(\theta)Z^{-1}(\theta') \begin{pmatrix} g_{1n}(\theta') \\ g_{2n}(\theta') \end{pmatrix} d\theta'. \end{aligned} \tag{4.16}$$

So, complete solution of the equation of motion for second order is written as

$$\begin{aligned} \begin{pmatrix} x(\theta) \\ p_x(\theta) \end{pmatrix} &= X(\theta)\begin{pmatrix} x_0 \\ p_{x0} \end{pmatrix} + A_1 x_0^2 + A_2 x_0 p_{x0} + A_3 p_{x0}^2 + A_4 z_0^2 + A_5 z_0 p_{z0} + A_6 p_{z0}^2, \\ \begin{pmatrix} z(\theta) \\ p_z(\theta) \end{pmatrix} &= Z(\theta)\begin{pmatrix} z_0 \\ p_{z0} \end{pmatrix} + B_1 x_0 z_0 + B_2 x_0 p_{z0} + B_3 p_{x0} z_0 + B_4 p_{x0} p_{z0}. \end{aligned} \tag{4.17}$$

With Eq. (4.17), we can track the equations of motion of beam extended to the second order. These equations of motion can be used to calculate the non-linear dynamics of the beam. In the next section, we will study the algorithm of the second order code applied to get the solution.

4.2 Second order code algorithm

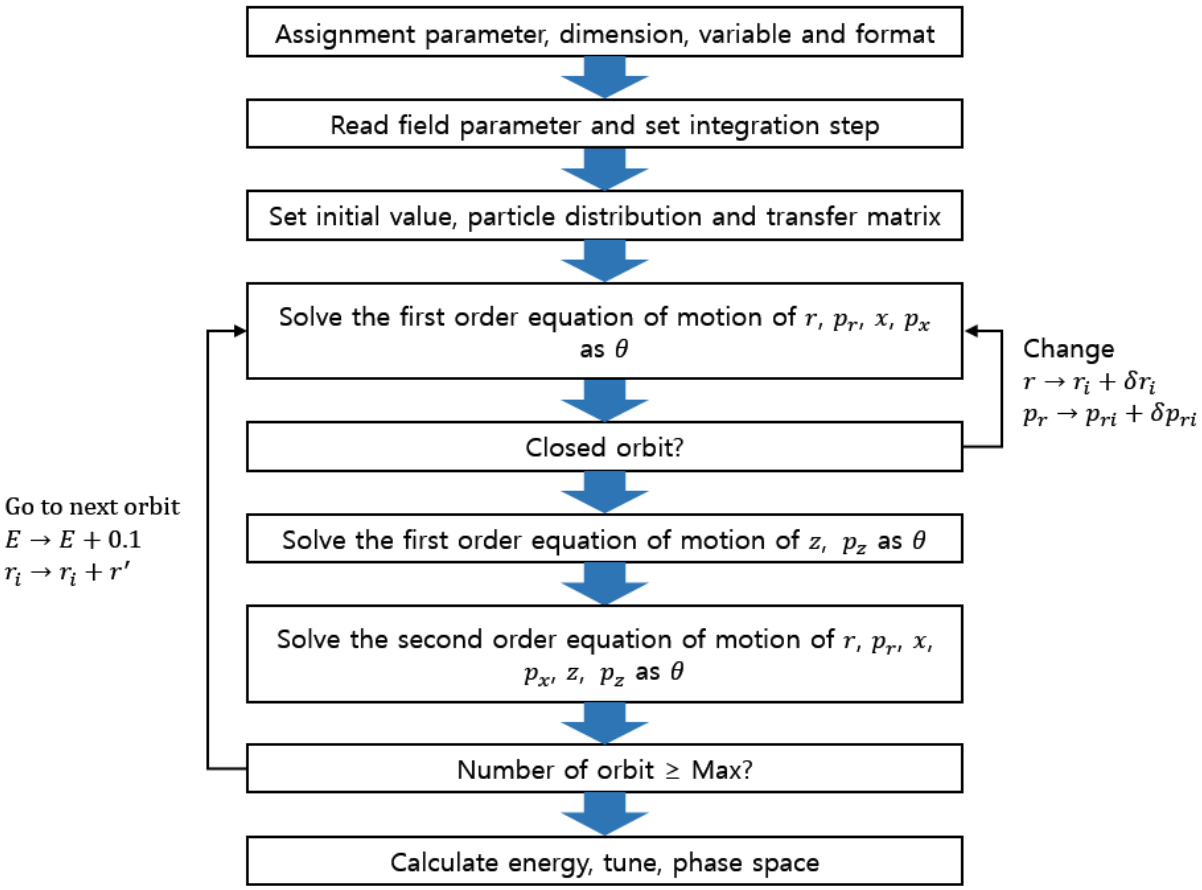


Fig. 4.1. Algorithm of second order code

The algorithm of the second order code is similar to the E.O code. The second order terms are added to the existing E.O code. First, we input magnetic field data of cyclotron and set integration step, initial values and initial particles distribution. Then the code computes only the horizontal direction of the first order equation of motion, and finds the closed orbit with this computation. If the code finds a closed orbit, the code tracks the particles with the second order equation of motion using the initial position and momentum with which the closed orbit was found. Finally, after tracking the beam to maximum energy, the code computes the distribution of the beam at each closed orbit point.

4.3 Full equation of motion code

The F.E code (full equation of motion code) tracks particles with given equations of motion. This code tracks each particle without expanding the equation like E.O code and second order code. So, computation time of the F.E code is slow compared to the other codes, but computation is accurate. The equations of motion of particles in F.E code can be written as

$$\frac{dr}{d\theta} = \frac{rp_r}{p_\theta}, \tag{4.18}$$

$$\frac{dp_r}{d\theta} = p_\theta - r\left(B - \frac{1}{2}z^2\left(\vec{\nabla}_\perp^2 B\right)\right) + \frac{zp_z}{p_\theta}\frac{\partial B}{\partial \theta}, \tag{4.19}$$

$$\frac{dz}{d\theta} = \frac{rp_z}{p_\theta}, \tag{4.20}$$

$$\frac{dp_z}{d\theta} = z\left(r\frac{\partial B}{\partial r} - \frac{p_r}{p_\theta}\frac{\partial B}{\partial \theta}\right). \tag{4.21}$$

In the F.E code, the horizontal components x and p_x are written as

$$x = r_{full} - r_{first}, \tag{4.22}$$

$$p_x = p_{r_{full}} - p_{r_{first}}, \tag{4.23}$$

where r_{full} and $p_{r_{full}}$ are values computed by Eqs. (4.18) and (4.19), and r_{first} and $p_{r_{first}}$ are values computed by Eqs. (3.9) and (3.10) in E.O code. Therefore, we obtain the horizontal components x and p_x in the F.E code, and can compute the horizontal and vertical particle distributions.

In the next section, we will compare the particle distributions, which are the results of simulation of each code.

4.4 Result of simulation

Due to the non-linear forces, phase space of the real beam may not be simply ellipses. Non-linear forces are induced by non-linear magnetic fields, and increase the RMS emittance which is the area occupied by the beam in phase space. However, in the E.O code by using a linear transfer matrix, the phase space of the beam proceeds with an ellipse (Fig. 3.8). Therefore, we will compare the results of E.O code with the second order code and F.E code containing all the non-linear terms.

Using KIRAMS-13 field data, we simulated the particle distributions with each code, which are E.O code, second order code and F.E code. We set the initial values extremely to compare the simulation values of each code. As shown in Fig. 4.2, we made the initial particle distribution a nearly straight line, not an ellipse. We set 5000 particles to Gaussian distribution with $\sigma_x = 0.05\ mm$, $\sigma_{p_x} = 0.0001\ mrad$, $\sigma_z = 0.05\ mm$ and $\sigma_{p_z} = 0.0001\ mrad$, where σ is standard deviation.

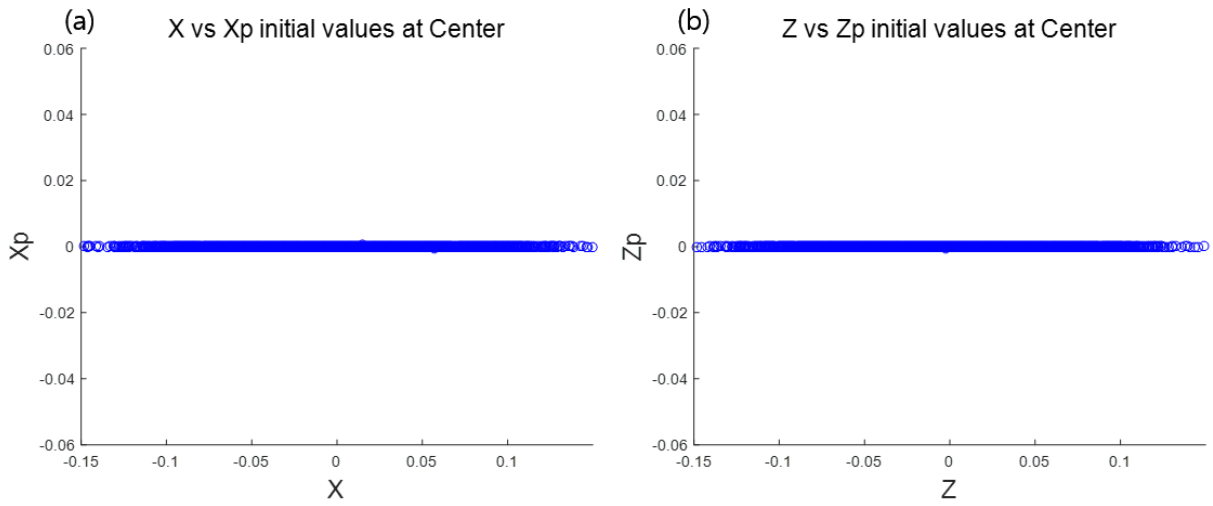


Fig. 4.2. Phase space of initial particles. In Fig.4.2 (a), graph of standard deviation is given by $\boldsymbol{\sigma_x = 0.05\ mm, \sigma_{p_x} = 0.0001\ mrad}$. In Fig. 4.2 (b), graph of standard deviation is given by $\boldsymbol{\sigma_z = 0.05\ mm, \sigma_{p_z} = 0.0001\ mrad}$. The number of particle is 5000 and particles are Gaussian distributions.

Figure 4.3 shows the results of the particle distributions calculated from each code. The particle distribution is linear in the E.O code which uses only in the first order terms. However, Figs. 4.3(e) and 4.3(f), which are the results of simulation using F.E code describing the actual particle motion, show that the particle distribution becomes a non-linear. Figures 4.3(c) and 4.3(d) show the results of simulation using the second order code. The second order code with non-linear terms shows that the particle distribution is not linear and is similar to the particle distribution in F.E code.

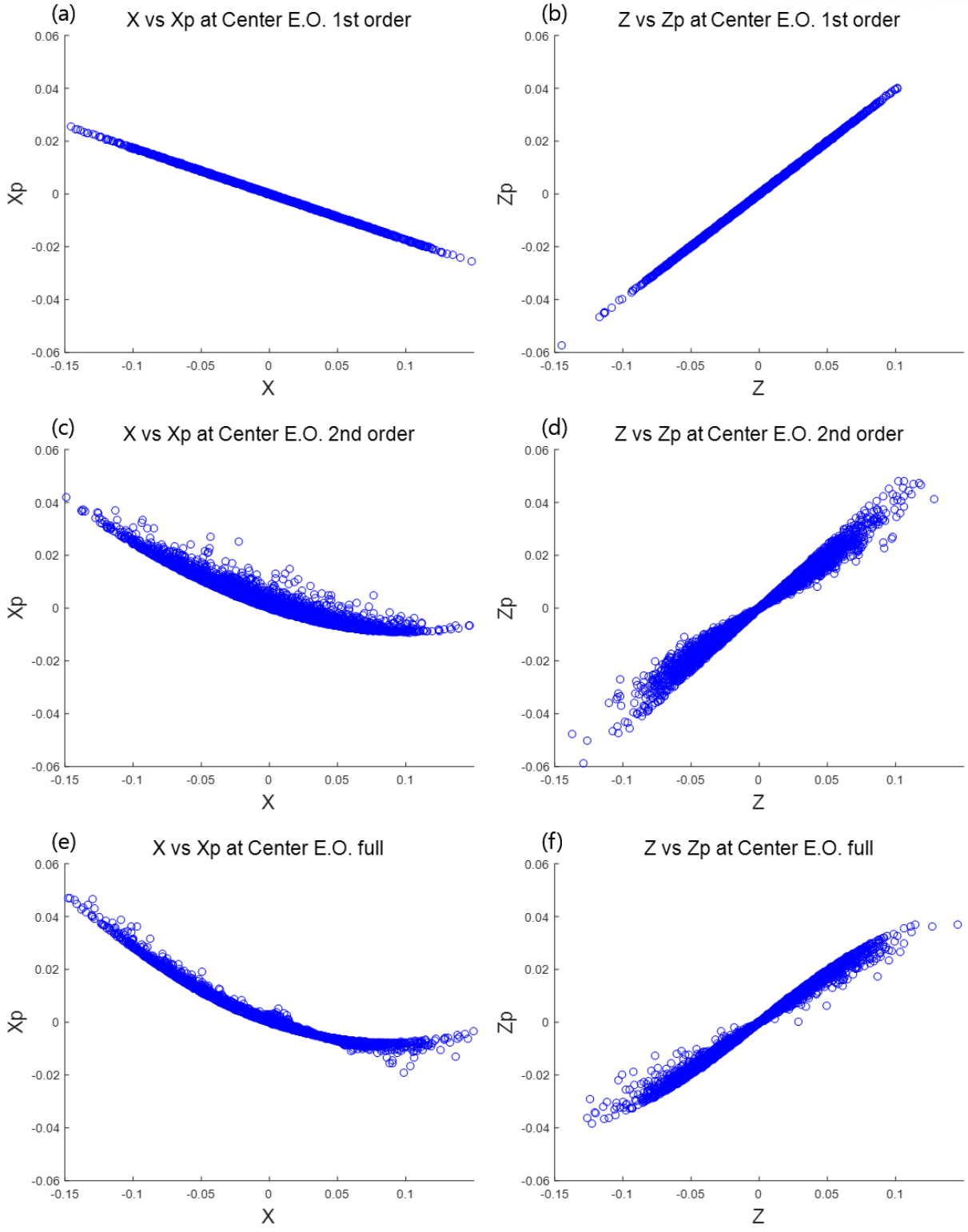


Fig. 4.3. Results of simulation of each code at 123 turns (12.61 MeV). We use KIRAMS-13 field data. The initial particle distribution is given by $\boldsymbol{\sigma_x = 0.05\ mm, \sigma_{p_x} = 0.0001\ mrad, \sigma_z = 0.05\ mm}$ and $\boldsymbol{\sigma_{p_z} = 0.0001\ mrad}$, and the number of the particle is 5000. Plots (a) and (b) graph are particle distributions from E.O code, (c) and (d) from second order code, and (e) and (f) from F.E code. Plots (a), (c) and (e) are particle distributions of horizontal direction, and (b), (d) and (f) are of vertical direction.

Figure 4.3 shows that the particle distribution tracking the particles in the second order terms is similar to in the full equation of motion. We can analyze each code by calculating RMS emittance of beam in addition to the comparison of Fig. 4.3 above. The RMS emittance can be obtained as

$$\epsilon_{rms} = \sqrt{\langle x^2 \rangle \langle x'^2 \rangle - \langle x x' \rangle^2}. \tag{4.24}$$

The RMS emittance results of each code using Eq. (4.24) are shown in Table. 4.1.

	Horizontal direction	Vertical direction
RMS emittance of initial distribution	$4.98 \times 10^{-6} \mu m^2$	$4.94 \times 10^{-6} \mu m^2$
RMS emittance results using E.O code	$4.98 \times 10^{-6} \mu m^2$	$4.94 \times 10^{-6} \mu m^2$
RMS emittance results using second order code	$1.70 \times 10^{-4} \mu m^2$	$6.34 \times 10^{-5} \mu m^2$
RMS emittance results using F.E code	$1.61 \times 10^{-4} \mu m^2$	$5.20 \times 10^{-5} \mu m^2$

Table 4.1. The RMS emittance results of each code in Fig. 4.3

The difference between the RMS emittance results of the E.O code and the second order code is 20 times because we set the initial condition to be almost a line rather than an ellipse as shown in Fig. 4.2. Because E.O code is calculated with matrix using first order terms, emittance result of the E.O code is same as the emittance of initial distribution. And we can see that the RMS emittance results of the second order code are similar to the F.E code results.

However, these two codes differ in computation time. When tracking 5000 particles to energy of 12.71 MeV (124 turns), the E.O code, the second order and F.E code took 1 second, 8 seconds and 7 minutes and 38 seconds of calculation time, respectively. As a result, we find that the calculation time of the second order code is 57 times faster than the calculation time of the F.E code.

Since the output current of KIRAMS-13 is $80\ \mu$A, the number of particles is about 5×10^{12}. To simulate many particles with F.E code, it takes a lot of computation time and computation efficiency is low. Therefore, it is faster to analyze the motion of particles using second order code. Also, the second order code is more computationally efficient because high-power cyclotron outputs many particles.

The second order terms in Eqs. (4.1)-(4.4) depend on the amount of change in the magnetic field. Therefore, it is expected that the effect of second order terms will be greater in high-power cyclotrons where the change of magnetic field is larger. So, we simulated with the field data of 430 MeV cyclotron.

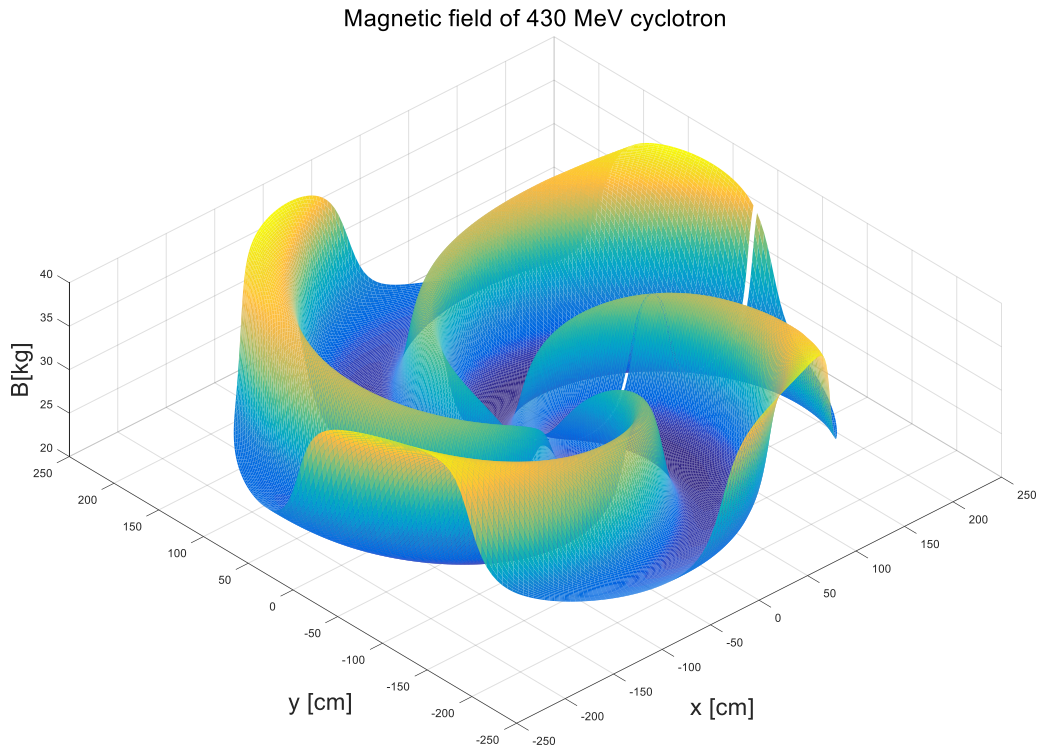


Fig. 4.4. Magnetic field data of 430 MeV cyclotron [14]

Figure 4.4 is the field data plot of 430 MeV cyclotron. It is composed of four sectors of spiral shape, and the difference between hill and valley is about 17~18 KG. This is about 5 KG larger than the hill valley difference in KIRAMS-13 (see Fig. 2.2). Therefore, the effect of the second order terms on the 430 MeV cyclotron is expected to be more significant than with the field data of KIRAMS-13.

Using 430 MeV cyclotron field data, we simulated the particles distributions with each code. As shown in Fig. 4.5, we made the initial particle distribution a nearly straight line. We set 5000 particles to Gaussian distribution with $\sigma_x = 0.2\,mm$, $\sigma_{p_x} = 0.0001\,mard$, $\sigma_z = 0.2\,mm$ and $\sigma_{p_z} = 0.0001\,mrad$, where σ is standard deviation.

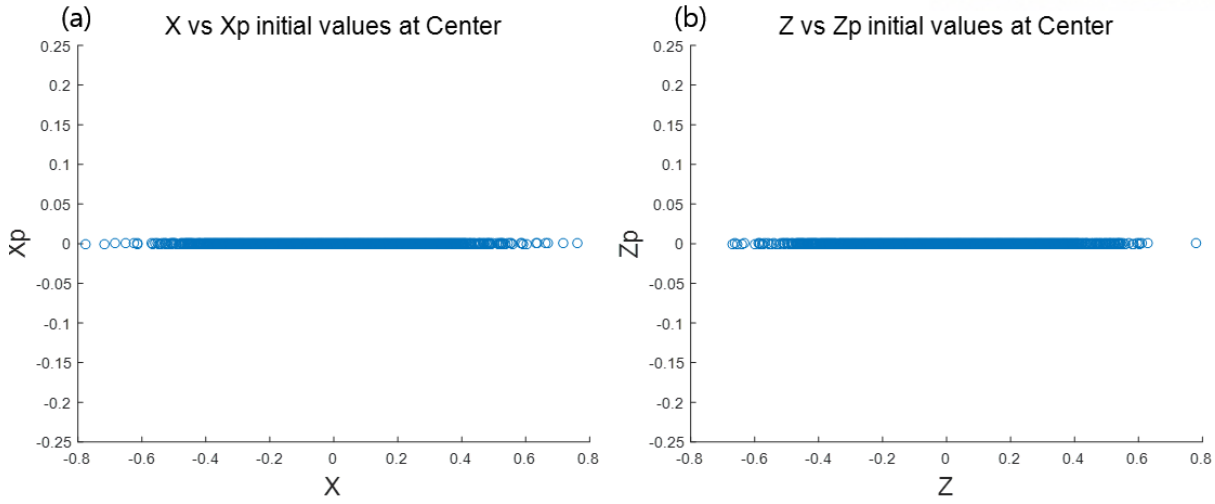


Fig. 4.5. Phase space of initial particles. In Fig. 4.5 (a), graph of standard deviation is given by $\boldsymbol{\sigma_x = 0.2\ mm, \sigma_{p_x} = 0.0001\ mrad}$. In Fig. 4.5 (b), graph of standard deviation is given by $\boldsymbol{\sigma_z = 0.2\ mm, \sigma_{p_z} = 0.0001\ mrad}$. The number of particle is 5000 and particles are Gaussian distributions.

Figures 4.6(c) and 4.6(d) show the results of simulation using the second order code. The second order code with non-linear terms shows that the particle distribution is not linear and is similar to the particle distribution in F.E code. The second order code shows a significant difference in particle distribution results compared to E.O code. When tracking 5000 particles to energy of 320 MeV (300 turns), the second order and F.E code took 20 seconds and 20 minutes of calculation time, respectively. As the result, we conclude that the second order code is more accurate and computationally efficient when simulating the high-power cyclotrons. Thus, the second order code will be an effective tool for designing high-power cyclotrons.

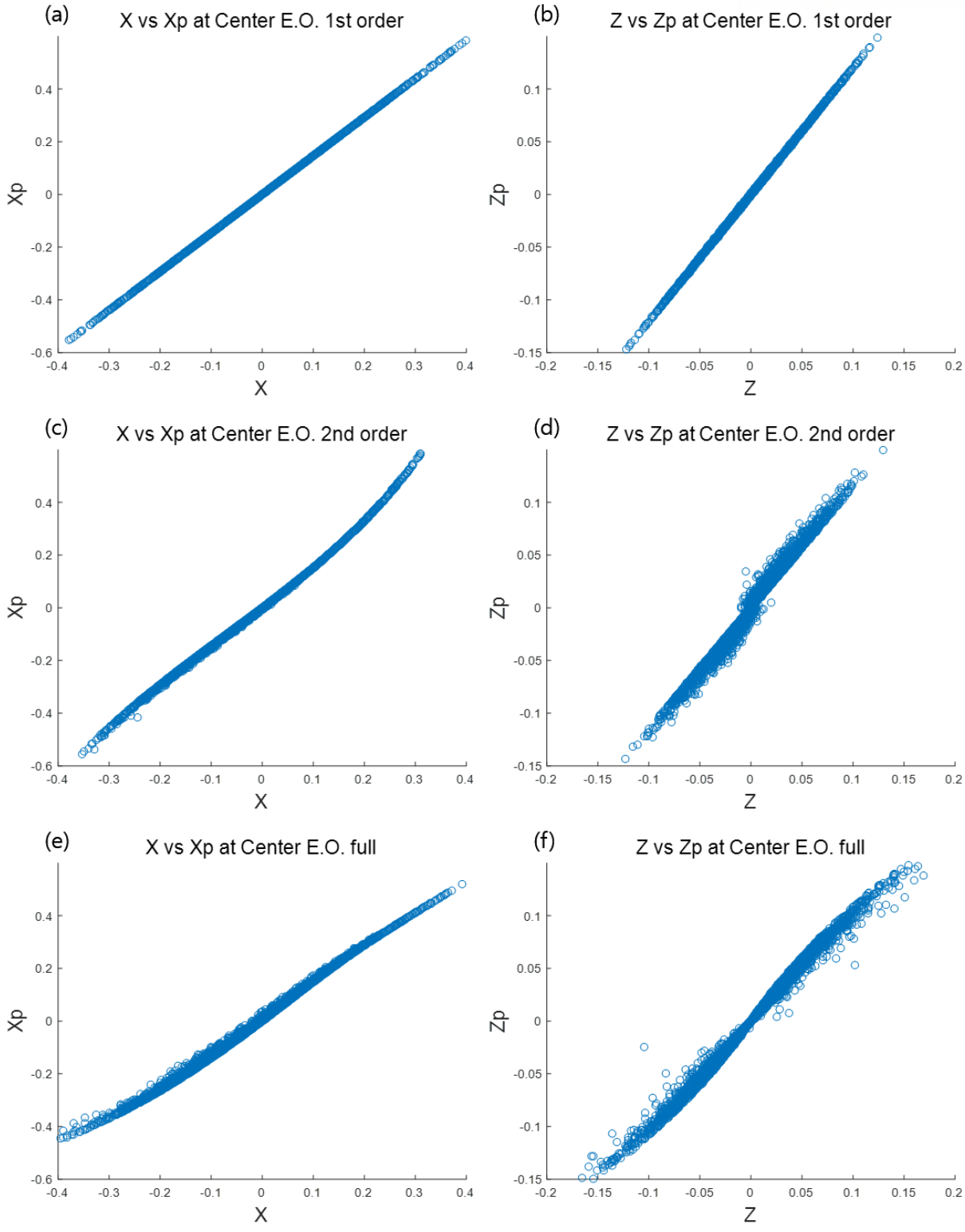


Fig. 4.6. Result of simulation of each code at 300 turns (320 MeV). we use 430MeV cyclotron field data. The initial particle distribution is given by $\boldsymbol{\sigma_x = 0.2\ mm, \sigma_{p_x} = 0.0001\ mrad}$, $\boldsymbol{\sigma_z = 0.2\ mm}$ and $\boldsymbol{\sigma_{p_z} = 0.0001\ mrad}$, and number of the particle is 5000. Plots (a) and (b) graph are particle distributions from E.O code, (c) and (d) from second order code, and (e) and (f) from F.E code. Plots (a), (c) and (e) are particle distribution of horizontal direction, and (b), (d) and (f) are of vertical direction.

Chapter 5. Effects of Acceleration

The existing E.O code only considered the magnet without considering the RF cavity. In other words, the E.O code does not consider the electric field. If a closed orbit is found in one energy orbit, the code adds arbitrary energy to the particles and repeats to search a new closed orbit of next energy. Therefore, by analyzing the tune and phase space of the beam in each energy trajectory, we analyze whether the beam is stable at the corresponding energy.

On the other hand, tracking the beam by adding electric field terms can reveal a continuous beam orbit. In the cyclotrons, it is important to observe the beam's orbit in the central region, which is a low-energy region where the effect of the electric field is greater than the magnetic field. Besides, we should confirm the continuous trajectory of the beam near the extraction, and analyze that the beam is extracted without loss. Therefore, it is necessary to develop a code that includes an electric field.

There are two types of acceleration methods that are implemented in the code that considers the acceleration effect. The first is to accelerate the particle by putting electric field data in Matlab, and the second is to accelerate the particles by delta function of the electric field.

The first method extracts the electric field data for each mesh in the real RF cavity and puts them into the code. So, this method can provide the exact beam trajectory, but the code execution is slow. Therefore, it is useful in the central region of cyclotrons, which has complex structure.

The second method speeds up code execution because it tracks the beam by defining the electric field as a delta function without data. This is fastest part and is used from about turn 5 to extraction, which is the area where we want to know roughly the trajectory of the beam.

We considered the acceleration effect by defining the electric field as a delta function, and do not consider electric field data.

5.1 Acceleration code algorithm

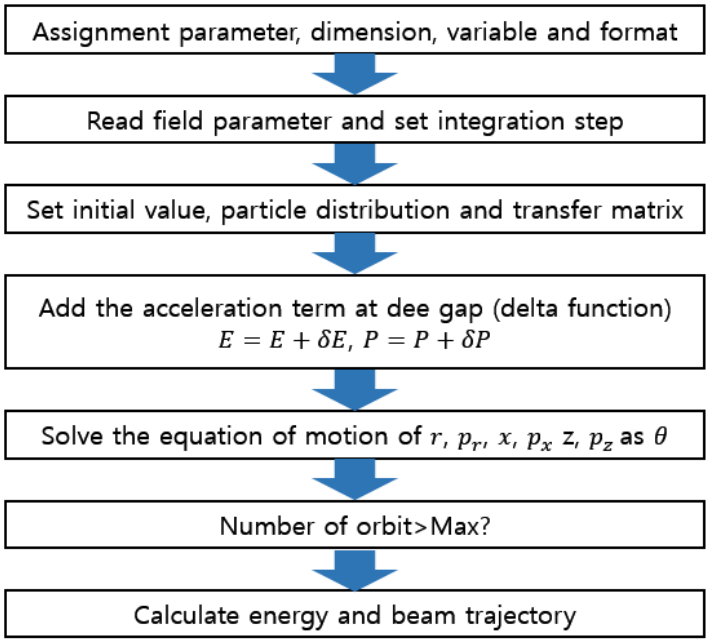


Fig. 5.1. Algorithm of acceleration code

The algorithm of the acceleration code is shown in Fig. 5.1. The acceleration code algorithm is simpler than the previous E.O code. Acceleration code does not correct for numerical errors. In other words, the end of one orbit is taken as the starting point of the next orbit and a continuous orbit is shown. So, the E.O code does not solve the numerical error for each orbit and traces the particles continuously to the extraction energy.

Therefore, the initial value in the acceleration code should not be arbitrarily set. If we set the first value arbitrary, the particles will diverge, or the code will stop. To set initial value in acceleration code, the initial position, momentum, and energy for each orbit are extracted from the E.O code and input into the initial value of the acceleration code.

Energy [MeV]	Radius [inch]	Momentum of R [mrad]	Energy [MeV]	Radius [cm]	Momentum of R [mrad]
0.31	2.487	0.05003	1.81	6.198	-0.01311
0.41	2.877	-0.00263	1.91	6.366	-0.01663
0.51	3.257	0.04529	2.01	6.528	-0.01572
0.61	3.586	0.07955	2.11	6.685	-0.00845
0.71	3.850	0.03643	2.21	6.831	-0.00603
0.81	4.122	0.02631	2.31	6.977	-0.01758
0.91	4.374	0.04955	2.41	7.121	-0.02355
1.01	4.610	0.00663	2.51	7.269	-0.02938
1.11	4.834	-0.00858	2.61	7.422	-0.03710
1.21	5.049	-0.01170	2.71	7.574	-0.04326
1.31	5.247	0.01532	2.81	7.718	-0.04496
1.41	5.450	-0.01376	2.91	7.857	-0.04373
1.51	5.649	-0.01642	3.01	7.990	-0.04106
1.61	5.840	-0.00853	3.11	8.121	-0.03910
1.71	6.023	-0.00688	3.21	8.249	-0.03853

Table 5.1. The initial position and momentum at each energy in KIRAMS-13 cyclotron

Table 5.1 shows the initial position and momentum at each energy of KIRAMS-13 extracted from E.O code. Referring to Table 5.1, we input the above values into the acceleration code for simulations. Since we made the electric field a delta function, we have input the initial position and momentum at 1.11 MeV, not the low-energy region.

5.2 Acceleration method

The equations of motion of the particles [15] passing through the electric field in the cyclotron are given by

$$p_\theta = \sqrt{p^2 - p_r^2 - p_z^2}, \tag{5.1}$$

$$\frac{dr}{d\theta} = \frac{rp_r}{\sqrt{p^2 - p_r^2 - p_z^2}}, \ \frac{dz}{d\theta} = \frac{rz}{\sqrt{p^2 - p_r^2 - p_z^2}}, \tag{5.2}$$

$$\frac{dp_r}{d\theta} = \sqrt{p^2 - p_r^2 - p_z^2} - qrB(r,\theta) + qt'\varepsilon_r, \ \frac{dp_z}{d\theta} = q\left[r\frac{\partial B}{\partial r} - \frac{p_r}{p_\theta}\frac{\partial B}{\partial \theta}\right]z + qt'\varepsilon_z, \tag{5.3}$$

$$\frac{dt}{d\theta} = \frac{\gamma m_o r}{\sqrt{p^2 - p_r^2 - p_z^2}}, \tag{5.4}$$

$$\frac{dE}{d\theta} = q\frac{r}{p_\theta}(p_r\varepsilon_r + p_\theta\varepsilon_\theta + p_z\varepsilon_z), \tag{5.5}$$

where; ε is the electric field, t' is $\frac{dt}{d\theta}$.

Equations (5.1)-(5.5) are similar to the equations of motion of the particles in a magnetic field, but electric field terms is added in the differential equations of momentum and energy. Also, these equations are the equations of motion when using electric field data. The equations of motion applied in this code with the electric field as a delta function are written as

$$p_\theta = \sqrt{p^2 - p_r^2 - p_z^2}, \tag{5.6}$$

$$\frac{dr}{d\theta} = \frac{rp_r}{\sqrt{p^2 - p_r^2 - p_z^2}}, \ \frac{dz}{d\theta} = \frac{rz}{\sqrt{p^2 - p_r^2 - p_z^2}}, \tag{5.7}$$

$$\frac{dp_r}{d\theta} = \sqrt{p^2 - p_r^2 - p_z^2} - qrB(r,\theta), \ \frac{dp_z}{d\theta} = q\left[r\frac{\partial B}{\partial r} - \frac{p_r}{p_\theta}\frac{\partial B}{\partial \theta}\right]z, \tag{5.8}$$

$$\frac{dE}{d\theta} = q\frac{r}{p_\theta}p_\theta\varepsilon_\theta = qr\varepsilon_\theta. \tag{5.9}$$

In cyclotrons, the particles move and accelerate in theta direction. To make the electric field a delta function, we should simplify the electric field. Since the theta direction of the electric field is the largest, ε_r and ε_z which are radial and vertical components of electric field, are ignored. Also, since the phase shift is not applied in this code, the particles are always set to get the maximum value of the RF voltage. Therefore, Eq. (5.4) has not been applied in the code. Equation (5.9) shows that as the particles pass through the electric field, the energy of the particles increases, and the momentum p increases. The equation for momentum and energy is written as

$$p = \left[\frac{E}{mc^2}\left(2 + \frac{E}{mc^2}\right)\right]^{1/2}, \tag{5.10}$$

and increases in proportion to the energy increase.

RF frequency	77.3 MHz
Harmonic number	4
Number of Dees	2
Dee angular width	39°
Dee voltage	45 kV

Table 5.2. RF cavity specification of KIRAMS-13

Table 5.2 shows the RF cavity specifications of KIRAMS-13. The acceleration effect is considered by referring to the above table. The KIRAMS-13 has two Dees and the Dee angular width is 39°. However, we set the angular width to 40° in the code because the acceleration code tracks particles every 2°. Since there are two Dees, there are four Dee gaps where the particles accelerate. The Dee voltage is 45 kV and particles gain 45 keV of energy for every Dee gap. The approximate picture of Dee in cyclotron is shown in Fig. 5.2.

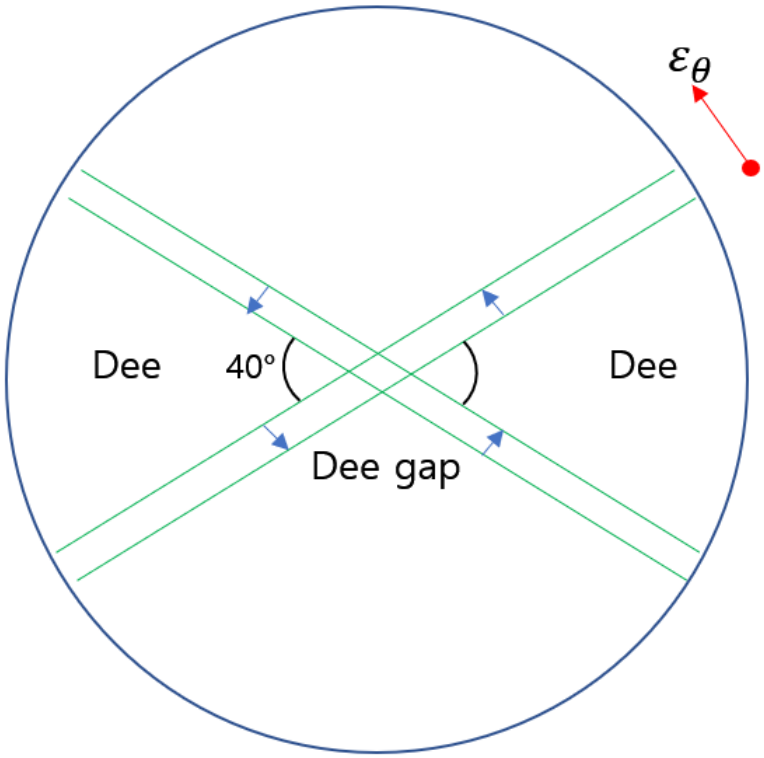


Fig. 5.2. The approximate picture of Dees at cyclotron

As Dee angular width is 40° as shown in the above picture, the effect of increasing the energy of particles is added at 20°, 160°, 200° and 340°.

The Dee gap in Fig. 5.2 is constant even with increasing radius. However, since the equation of

motion in the acceleration code is a differential equation for theta, the arc length (the length of the Dee gap) increases as the radius of the particle increases. And as the radius increases at each Dee gap, the effect of the electric field becomes larger and energy increases more. Therefore, to obtain a constant effect of the electric field at each radius, we divide ε_θ by the length of the arc as

$$\varepsilon_\theta \rightarrow \frac{\varepsilon_\theta}{2\pi r/360^\circ}. \tag{5.11}$$

Therefore, we input 45 kV for ε_θ and divide by the length of the arc.

5.3 Result of simulation

We simulated using field data from KIRAMS-13 [16]. Also, referring to the RF cavity specification of KIRAMS-13, the particles were accelerated at 20°, 160°, 200° and 340° with 45 kV.

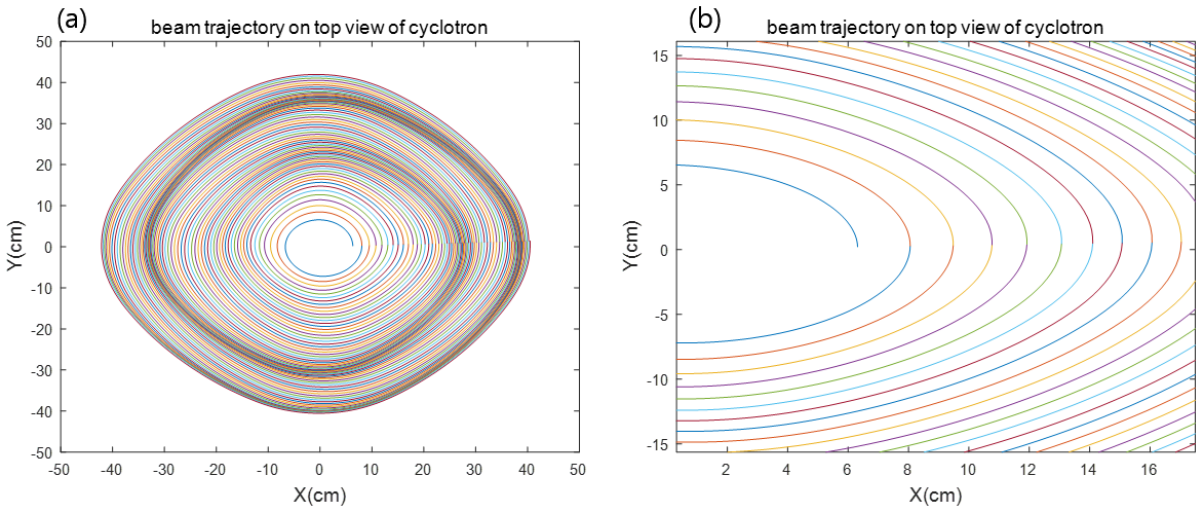


Fig. 5.3. (a) Beam trajectory on top view of cyclotron and (b) enlarged view of the center of (a)

As shown in Fig. 5.3, we tracked a continuous beam orbit up to 13 MeV. The beam gets energy from the four Dee gap regions and the radius becomes large. Therefore, orbit of the beam is a spiral.

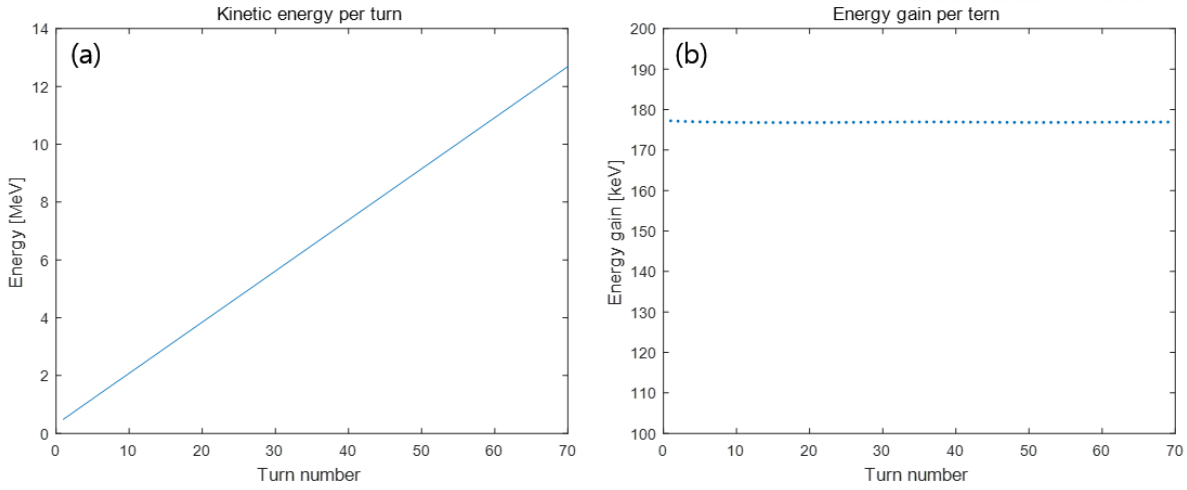


Fig. 5.4. (a) Kinetic energy per turn, and (b) energy gain per tern

Fig. 5.4 (a) shows the energy according to turn. The energy seems to increase linearly. However, (b) graph shows that the energy gain per turn is slightly different. The energy increases by about 176.9 keV per turn. Reason for difference of the energy gain per turn is due to numerical errors.

We modified the code to include acceleration effects. We could analyze the continuous beam trajectory and energy gain per turn. In the future, we plan to upgrade the acceleration code by inputting electric field data, and the upgraded code is expected to perform better for computing more accurate beam trajectory in the central region. The upgraded acceleration code is important for determining the orbit of the beam in the high-power cyclotrons. Because of the high beam current in the high-power cyclotrons, the space charge effect and the repulsive force of each beam bunch increases. Through the development of acceleration code, we can analyze the effect of each beam bunch.

Chapter 6. Summary and Future Work

In this thesis, we have studied beam dynamics in cyclotrons. Also, we have described the existing E.O code, the newly developed second order code, and the acceleration code. The second order terms can be obtained by linear expansion of the existing equation of motion of particles and the acceleration effect was generated by inputting the electric field as a delta function instead of the electric field.

The code with the second order terms added was faster than the code with the full equation. The second order code also tracked the particle trajectories more accurately than E.O code. Therefore, this code has advantages of accuracy and fast computation. With these advantages, it is possible to grasp magnet errors with more accurate and faster computation time by using second order code when developing high-power cyclotron magnets in the future. In addition, the acceleration code developed by inputting the electric field as a delta function enabled continuous beam tracking. By tracking the beam continuously, we can find the direction of the beam motion in cyclotron, and will be able to extract the beam properties more accurately in the future.

However, in the codes developed in this thesis, several important physical phenomena are still missing. First, we need to create electric field data to make more accurate acceleration effect. Second, we need to consider the voltage applied to the particles when the RF frequency and the angular frequency of the particles are different by adding a phase shift effect. Third, a more accurate beam dynamic for a high-power cyclotron should be implemented such as space charge effects and transverse to longitudinal coupling.

In the future, we will be able to design a high-power cyclotron for ADS based on more advanced simulation code.

7. Appendix

7.1 Taylor expansion to first order

We can expand the equation of motion as

$$\begin{aligned} r' &= r'_0 + r'_1 + r'_{x^2} + r'_{z^2}, \\ p'_r &= p'_{r_0} + p'_{r_1} + p'_{r_{x^2}} + p'_{r_{z^2}}, \\ z' &= z'_1 + z'_2, \\ p'_z &= p'_{z_1} + p'_{z_2}. \end{aligned} \tag{7.1}$$

r'_1 and p'_{r_1} are first order terms of horizontal components, and z'_1 and p'_{z_1} are first order terms of vertical components, and r'_2, p'_{r_2}, z'_2 and p'_{z_2} are second order terms. When expand coordinates such that $r \to r + x$, $p_r \to p_r + p_x$, we can derive the expanding equation such that

$$\begin{aligned} r'_1 &= \frac{\partial}{\partial x}\left(\frac{dr}{d\theta}\right)\Big|_{x,p_x,z,p_z=0} x + \frac{\partial}{\partial p_x}\left(\frac{dr}{d\theta}\right)\Big|_{x,p_x,z,p_z=0} p_x \\ &\quad + \frac{\partial}{\partial z}\left(\frac{dr}{d\theta}\right)\Big|_{x,p_x,z,p_z=0} z + \frac{\partial}{\partial p_z}\left(\frac{dr}{d\theta}\right)\Big|_{x,p_x,z,p_z=0} p_z, \\ r'_1 &= \frac{\partial}{\partial x}\frac{(r+x)(p_r+p_x)}{\sqrt{P^2-(p_r+p_x)^2-p_z^2}}\Big|_{x,p_x,z,p_z=0} x + \frac{\partial}{\partial p_x}\frac{(r+x)(p_r+p_x)}{\sqrt{P^2-(p_r+p_x)^2-p_z^2}}\Big|_{x,p_x,z,p_z=0} p_x \\ &\quad + \frac{\partial}{\partial p_z}\frac{(r+x)(p_r+p_x)}{\sqrt{P^2-(p_r+p_x)^2-p_z^2}}\Big|_{x,p_x,z,p_z=0} p_z, \\ r'_1 &= \frac{(p_r+p_x)}{\sqrt{P^2-(p_r+p_x)^2-p_z^2}}\Big|_{x,p_x,z,p_z=0} x \\ &\quad + \left(\frac{(r+x)}{\sqrt{P^2-(p_r+p_x)^2-p_z^2}} + \frac{(r+x)(p_r+p_x)(p_r+p_x)}{\sqrt{\left(P^2-(p_r+p_x)^2-p_z^2\right)^3}}\right)\Big|_{x,p_x,z,p_z=0} p_x \\ &\quad + \frac{(r+x)(p_r+p_x)p_z}{\sqrt{P^2-(p_r+p_x)^2-p_z^2}}\Big|_{x,p_x,z,p_z=0} p_z, \\ r'_1 &= \frac{p_r}{p_\theta}x + \left(\frac{rp_\theta^2}{p_\theta^3} + \frac{rp_r^2}{p_\theta^3}\right)p_x, \qquad \left(P^2 = p_r^2 + p_\theta^2\right) \\ r'_1 &= \frac{p_r}{p_\theta}x + \left(\frac{rP^2}{p_\theta^3}\right)p_x, \end{aligned} \tag{7.2}$$

$$p'_{r_1} = p_\theta - r\left(B - \frac{1}{2}z^2\left(\vec{\nabla}_\perp^2 B\right)\right) + \frac{\mathrm{zp_z}}{\sqrt{P^2 - p_r^2 - p_z^2}}\frac{\partial B}{\partial \theta},$$

$$p'_{r_1} = \frac{\partial}{\partial x}\left(\sqrt{P^2 - (p_r + p_x)^2 - p_z^2} - (r+x)\left(B - \frac{1}{2}z^2\left(\vec{\nabla}_\perp^2 B\right)\right) + \frac{\mathrm{zp_z}}{\sqrt{P^2 - (p_r+p_x)^2 - p_z^2}}\frac{\partial B}{\partial \theta}\right)\Bigg|_{x,p_x,z,p_z=0} x$$

$$+ \frac{\partial}{\partial p_x}\left(\sqrt{P^2 - (p_r + p_x)^2 - p_z^2} - (r+x)\left(B - \frac{1}{2}z^2\left(\vec{\nabla}_\perp^2 B\right)\right) + \frac{zp_z}{\sqrt{P^2 - (p_r+p_x)^2 - p_z^2}}\frac{\partial B}{\partial \theta}\right)\Bigg|_{x,p_x,z,p_z=0} p_x \quad (7.3)$$

$$p'_{r_1} = -B - (r+x)\frac{\partial B}{\partial r}\Big|_{x,p_x,z,p_z=0} x + \left(\frac{-(p_r+p_x)}{\sqrt{P^2 - (p_r+p_x)^2 - p_z^2}}\right)\Bigg|_{x,p_x,z,p_z=0} p_x$$

$$p'_{r_1} = -\frac{p_r}{p_\theta}p_x - \left(B + r\frac{\partial B}{\partial r}\right)x,$$

$$z'_1 = \frac{\partial}{\partial x}\left(\frac{dz}{d\theta}\right)\Big|_{x,p_x,z,p_z=0} x + \frac{\partial}{\partial p_x}\left(\frac{dz}{d\theta}\right)\Big|_{x,p_x,z,p_z=0} p_x + \frac{\partial}{\partial z}\left(\frac{dz}{d\theta}\right)\Big|_{x,p_x,z,p_z=0} z$$

$$+ \frac{\partial}{\partial p_z}\left(\frac{dz}{d\theta}\right)\Big|_{x,p_x,z,p_z=0} p_z$$

$$z'_1 = \frac{\partial}{\partial x}\left(\frac{(r+x)p_z}{\sqrt{P^2 - (p_r+p_x)^2 - p_z^2}}\right)\Bigg|_{x,p_x,z,p_z=0} x + \frac{\partial}{\partial p_x}\left(\frac{(r+x)p_z}{\sqrt{P^2 - (p_r+p_x)^2 - p_z^2}}\right)\Bigg|_{x,p_x,z,p_z=0} p_x$$

$$+ \frac{\partial}{\partial z}\left(\frac{(r+x)p_z}{\sqrt{P^2 - (p_r+p_x)^2 - p_z^2}}\right)\Bigg|_{x,p_x,z,p_z=0} z + \frac{\partial}{\partial p_z}\left(\frac{(r+x)p_z}{\sqrt{P^2 - (p_r+p_x)^2 - p_z^2}}\right)\Bigg|_{x,p_x,z,p_z=0} p_z \quad (7.4)$$

$$z'_1 = \left(\frac{(r+x)}{\sqrt{P^2 - (p_r+p_x)^2 - p_z^2}}\right)\Bigg|_{x,p_x,z,p_z=0} p_z + \left(\frac{(r+x)p_z^2}{\sqrt{\left(P^2 - (p_r+p_x)^2 - p_z^2\right)^3}}\right)\Bigg|_{x,p_x,z,p_z=0} p_z$$

$$z'_1 = \frac{r}{p_\theta}p_z,$$

$$p'_{z_1} = \frac{\partial}{\partial x}\left(\frac{dp_z}{d\theta}\right)\Big|_{x,p_x,z,p_z=0} x + \frac{\partial}{\partial p_x}\left(\frac{dp_z}{d\theta}\right)\Big|_{x,p_x,z,p_z=0} p_x + \frac{\partial}{\partial z}\left(\frac{dp_z}{d\theta}\right)\Big|_{x,p_x,z,p_z=0} z$$

$$+ \frac{\partial}{\partial p_z}\left(\frac{dp_z}{d\theta}\right)\Big|_{x,p_x,z,p_z=0} p_z$$

$$p'_{z_1} = \frac{\partial}{\partial z}\left((r+x)z\frac{\partial B}{\partial r} - \frac{z(p_r+p_x)}{\sqrt{P^2 - (p_r+p_x)^2 - p_z^2}}\frac{\partial B}{\partial \theta}\right)\Bigg|_{x,p_x,z,p_z=0} z \quad (7.5)$$

$$p'_{z_1} = \left(r\frac{\partial B}{\partial r} - \frac{p_r}{p_\theta}\frac{\partial B}{\partial \theta}\right)z.$$

So, we can write the first order equation of motion as

$$\frac{dx}{d\theta} = \frac{p_r}{p_\theta}x + \left(\frac{rP^2}{p_\theta^3}\right)p_x, \quad \frac{dp_x}{d\theta} = -\frac{p_r}{p_\theta}p_x - \left(B + r\frac{\partial B}{\partial r}\right)x,$$

$$\frac{dz}{d\theta} = \frac{r}{p_\theta}p_z, \qquad \frac{dp_z}{d\theta} = \left(r\frac{\partial B}{\partial r} - \frac{p_r}{p_\theta}\frac{\partial B}{\partial \theta}\right)z. \quad (7.6)$$

7.2 Taylor expansion to second order

We can expand the equation of motion to second order terms (r_2', p_{r_2}', z_2' and p_{z_2}'). r_2' term can be derived such that

$$r'_{x^2} = \frac{1}{2}\frac{\partial^2}{\partial x^2}\left(\frac{dr}{d\theta}\right)\Big|_{x,p_x,z,p_z=0} x^2 + \frac{\partial^2}{\partial p_x \partial x}\left(\frac{dr}{d\theta}\right)\Big|_{x,p_x,z,p_z=0} xp_x + \frac{1}{2}\frac{\partial^2}{\partial p_x^2}\left(\frac{dr}{d\theta}\right)\Big|_{x,p_x,z,p_z=0} p_x^2, \tag{7.7}$$

$$\frac{\partial^2}{\partial x^2}\left(\frac{dr}{d\theta}\right)\Big|_{x,p_x,z,p_z=0} = \frac{\partial^2}{\partial x^2}\frac{(r+x)(p_r+p_x)}{\sqrt{P^2-(p_r+p_x)^2-p_z^2}}\Big|_{x,p_x,z,p_z=0} = 0,$$

$$\begin{aligned}\frac{\partial^2}{\partial p_x \partial x}\left(\frac{dr}{d\theta}\right)\Big|_{x,p_x,z,p_z=0} &= \frac{\partial^2}{\partial p_x \partial x}\frac{(r+x)(p_r+p_x)}{\sqrt{P^2-(p_r+p_x)^2-p_z^2}}\Big|_{x,p_x,z,p_z=0} \\ &= \frac{\partial}{\partial p_x}\frac{(p_r+p_x)}{\sqrt{P^2-(p_r+p_x)^2-p_z^2}}\Big|_{x,p_x,z,p_z=0} \\ &= \frac{1}{p_\theta} + \frac{p_r^2}{p_\theta^3} = \frac{p_\theta^2+p_r^2}{p_\theta^3} = \frac{P^2}{p_\theta^3},\end{aligned} \tag{7.8}$$

$$\frac{\partial^2}{\partial p_x^2}\frac{(r+x)(p_r+p_x)}{\sqrt{P^2-(p_r+p_x)^2-p_z^2}}\Big|_{x,p_x,z,p_z=0} = \frac{\partial}{\partial p_x}\left(\frac{(r+x)}{\sqrt{P^2-(p_r+p_x)^2-p_z^2}} + \frac{(r+x)(p_r+p_x)^2}{\sqrt{\left(P^2-(p_r+p_x)^2-p_z^2\right)^3}}\right)\Big|_{x,p_x,z,p_z=0}$$

$$= \left(\frac{(r+x)(p_r+p_x)}{\sqrt{\left(P^2-(p_r+p_x)^2-p_z^2\right)^3}} + \frac{2(r+x)(p_r+p_x)}{\sqrt{\left(P^2-(p_r+p_x)^2-p_z^2\right)^3}} + \frac{3(r+x)(p_r+p_x)^3}{\sqrt{\left(P^2-(p_r+p_x)^2-p_z^2\right)^5}}\right)\Big|_{x,p_x,z,p_z=0}$$

$$= \frac{rp_r}{p_\theta^3} + 2\frac{rp_r}{p_\theta^3} + 3\frac{rp_r^3}{p_\theta^5} = 3\frac{\mathrm{rp_r}(p_\theta^2+p_r^2)}{p_\theta^5} = 3\frac{rp_rP^2}{p_\theta^5},$$

$$r'_{x^2} = \frac{\partial^2}{\partial p_x \partial x}\left(\frac{dr}{d\theta}\right)\Big|_{x,p_x,z,p_z=0} xp_x + \frac{1}{2}\frac{\partial^2}{\partial p_x^2}\left(\frac{dr}{d\theta}\right)\Big|_{x,p_x,z,p_z=0} p_x^2 = \frac{P^2}{p_\theta^3}xp_x + \frac{3}{2}\frac{rp_rP^2}{p_\theta^5}p_x^2\,, \tag{7.9}$$

$$r'_{z^2} = \frac{1}{2}\frac{\partial^2}{\partial z^2}\left(\frac{dr}{d\theta}\right)\Big|_{x,p_x,z,p_z=0} z^2 + \frac{\partial^2}{\partial p_z \partial z}\left(\frac{dr}{d\theta}\right)\Big|_{x,p_x,z,p_z=0} zp_z + \frac{1}{2}\frac{\partial^2}{\partial p_z^2}\left(\frac{dr}{d\theta}\right)\Big|_{x,p_x,z,p_z=0} p_z^2, \quad (7.10)$$

$$\frac{\partial^2}{\partial z^2}\left(\frac{(r+x)(p_r+p_x)}{\sqrt{P^2-(p_r+p_x)^2-p_z^2}}\right)\Big|_{x,p_x,z,p_z=0} = 0,$$

$$\frac{\partial^2}{\partial p_z \partial z}\left(\frac{(r+x)(p_r+p_x)}{\sqrt{P^2-(p_r+p_x)^2-p_z^2}}\right)\Big|_{x,p_x,z,p_z=0} = 0,$$

$$\frac{\partial^2}{\partial p_z^2}\left(\frac{(r+x)(p_r+p_x)}{\sqrt{P^2-(p_r+p_x)^2-p_z^2}}\right)\Big|_{x,p_x,z,p_z=0} = \frac{\partial}{\partial p_z}\left(\frac{(r+x)(p_r+p_x)p_z}{\sqrt{\left(P^2-(p_r+p_x)^2-p_z^2\right)^3}}\right)\Big|_{x,p_x,z,p_z=0} \quad (7.11)$$

$$= \left(\frac{(r+x)(p_r+p_x)}{\sqrt{\left(P^2-(p_r+p_x)^2-p_z^2\right)^3}} + \frac{3(r+x)(p_r+p_x)p_z^2}{\sqrt{\left(P^2-(p_r+p_x)^2-p_z^2\right)^5}}\right)\Big|_{x,p_x,z,p_z=0} = \frac{rp_r}{p_\theta^3},$$

$$r'_{z^2} = \frac{1}{2}\frac{\partial^2}{\partial p_z^2}\left(\frac{dr}{d\theta}\right)\Big|_{x,p_x,z,p_z=0} p_z^2 = \frac{1}{2}\frac{rp_r}{p_\theta^3}p_z^2. \quad (7.12)$$

So r'_2 which is second order terms in horizontal radius is written as

$$r'_2 = \frac{P^2}{p_\theta^3}xp_x + \frac{3}{2}\frac{rp_rP^2}{p_\theta^5}p_x^2 + \frac{1}{2}\frac{rp_r}{p_\theta^3}p_z^2. \quad (7.13)$$

p'_{r_2} is derived such that

$$p'_{r_{x^2}} = \frac{1}{2}\frac{\partial^2}{\partial x^2}\left(\frac{dp_r}{d\theta}\right)\Big|_{x,p_x,z,p_z=0} x^2 + \frac{\partial^2}{\partial p_x \partial x}\left(\frac{dp_r}{d\theta}\right)\Big|_{x,p_x,z,p_z=0} xp_x + \frac{1}{2}\frac{\partial^2}{\partial p_x^2}\left(\frac{dp_r}{d\theta}\right)\Big|_{x,p_x,z,p_z=0} p_x^2, \quad (7.14)$$

$$\frac{\partial^2}{\partial x^2}\left(\sqrt{P^2-(p_r+p_x)^2-p_z^2} - (r+x)\left(B - \frac{1}{2}z^2\left(\vec{\nabla}_\perp^2 B\right)\right) + \frac{zp_z}{\sqrt{P^2-(p_r+p_x)^2-p_z^2}}\frac{\partial B}{\partial \theta}\right)\Big|_{x,p_x,z,p_z=0}$$

$$= \frac{\partial}{\partial x}\left(-B-(r+x)\frac{\partial B}{\partial x}\right)\Big|_{x,p_x,z,p_z=0} = \left(-\frac{\partial B}{\partial x} - \frac{\partial B}{\partial x} - (r+x)\frac{\partial^2 B}{\partial x^2}\right)\Big|_{x,p_x,z,p_z=0} \quad (7.15)$$

$$= -2\frac{\partial B}{\partial r} - r\frac{\partial^2 B}{\partial r^2},$$

$$\frac{\partial^2}{\partial p_x \partial x}\left(\sqrt{P^2-(p_r+p_x)^2-p_z^2}-(r+x)\left(B-\frac{1}{2}z^2\left(\vec{\nabla}_\perp^2 B\right)\right)+\frac{zp_z}{\sqrt{P^2-(p_r+p_x)^2-p_z^2}}\frac{\partial B}{\partial \theta}\right)\Bigg|_{x,p_x,z,p_z=0}$$

$$=\frac{\partial}{\partial p_x}\left(-B-(r+x)\frac{\partial B}{\partial x}\right)\Big|_{x,p_x,z,p_z=0}=0,$$

$$\frac{\partial^2}{\partial p_x^2}\left(\sqrt{P^2-(p_r+p_x)^2-p_z^2}-(r+x)\left(B-\frac{1}{2}z^2\left(\vec{\nabla}_\perp^2 B\right)\right)+\frac{zp_z}{\sqrt{P^2-(p_r+p_x)^2-p_z^2}}\frac{\partial B}{\partial \theta}\right)\Bigg|_{x,p_x,z,p_z=0}$$

$$=\frac{\partial}{\partial p_x}\left(\frac{-(p_r+p_x)}{\sqrt{P^2-\left(p_r+p_x\right)^2-p_z^2}}\right)\Bigg|_{x,p_x,z,p_z=0}=\left(\frac{-1}{\sqrt{P^2-\left(p_r+p_x\right)^2-p_z^2}}+\frac{\left(p_r+p_x\right)^2}{\sqrt{\left(P^2-\left(p_r+p_x\right)^2-p_z^2\right)^3}}\right)\Bigg|_{x,p_x,z,p_z=0},$$

$$=-\frac{1}{p_\theta}-\frac{p_r^2}{p_\theta^3}=-\frac{P^2}{p_\theta^3},$$

$$p'_{r_{x^2}}=-\frac{1}{2}\frac{P^2}{p_\theta^3}p_x^2-\frac{1}{2}\left(2\frac{\partial B}{\partial r}-r\frac{\partial^2 B}{\partial r^2}\right)x^2, \tag{7.16}$$

$$p'_{r_{z^2}}=\frac{1}{2}\frac{\partial^2}{\partial z^2}\left(\frac{dp_r}{d\theta}\right)\Big|_{x,p_x,z,p_z=0}z^2+\frac{\partial^2}{\partial p_z \partial z}\left(\frac{dp_r}{d\theta}\right)\Big|_{x,p_x,z,p_z=0}zp_z+\frac{1}{2}\frac{\partial^2}{\partial p_z^2}\left(\frac{dp_r}{d\theta}\right)\Big|_{x,p_x,z,p_z=0}p_z^2, \tag{7.17}$$

$$\frac{\partial^2}{\partial z^2}\left(\sqrt{P^2-\left(p_r+p_x\right)^2-p_z^2}-(r+x)\left(B-\frac{1}{2}z^2\left(\vec{\nabla}_\perp^2 B\right)\right)+\frac{zp_z}{\sqrt{P^2-\left(p_r+p_x\right)^2-p_z^2}}\frac{\partial B}{\partial \theta}\right)\Bigg|_{x,p_x,z,p_z=0}$$

$$=\frac{\partial}{\partial z}\left(-(r+x)\left(-z\left(\vec{\nabla}_\perp^2 B\right)\right)+\frac{p_z}{\sqrt{P^2-\left(p_r+p_x\right)^2-p_z^2}}\frac{\partial B}{\partial \theta}\right)\Bigg|_{x,p_x,z,p_z=0}=r\left(\vec{\nabla}_\perp^2 B\right),$$

$$\frac{\partial^2}{\partial p_z \partial z}\left(\sqrt{P^2-\left(p_r+p_x\right)^2-p_z^2}-(r+x)\left(B-\frac{1}{2}z^2\left(\vec{\nabla}_\perp^2 B\right)\right)+\frac{zp_z}{\sqrt{P^2-\left(p_r+p_x\right)^2-p_z^2}}\frac{\partial B}{\partial \theta}\right)\Bigg|_{x,p_x,z,p_z=0} \tag{7.18}$$

$$=\frac{\partial}{\partial p_z}\left(-(r+x)\left(-z\left(\vec{\nabla}_\perp^2 B\right)\right)+\frac{p_z}{\sqrt{P^2-\left(p_r+p_x\right)^2-p_z^2}}\frac{\partial B}{\partial \theta}\right)\Bigg|_{x,p_x,z,p_z=0}$$

$$=\left(\frac{1}{\sqrt{P^2-\left(p_r+p_x\right)^2-p_z^2}}-\frac{p_z^2}{\sqrt{\left(P^2-\left(p_r+p_x\right)^2-p_z^2\right)^3}}\right)\frac{\partial B}{\partial \theta}\Bigg|_{x,p_x,z,p_z=0}=\frac{\partial B}{\partial \theta}\frac{1}{p_\theta},$$

$$\frac{\partial^2}{\partial p_z^2}\left(\sqrt{P^2-\left(p_r+p_x\right)^2-p_z^2}-(r+x)\left(B-\frac{1}{2}z^2\left(\vec{\nabla}_\perp^2 B\right)\right)+\frac{zp_z}{\sqrt{P^2-\left(p_r+p_x\right)^2-p_z^2}}\frac{\partial B}{\partial\theta}\right)\Bigg|_{x,p_x,z,p_z=0}$$

$$\frac{\partial}{\partial p_z}\left(\frac{-p_z}{\sqrt{P^2-\left(p_r+p_x\right)^2-p_z^2}}+\frac{z}{\sqrt{P^2-\left(p_r+p_x\right)^2-p_z^2}}\frac{\partial B}{\partial\theta}\right)\Bigg|_{x,p_x,z,p_z=0}$$

$$=\left(\frac{-1}{\sqrt{P^2-\left(p_r+p_x\right)^2-p_z^2}}+\frac{-p_z}{\sqrt{\left(P^2-\left(p_r+p_x\right)^2-p_z^2\right)^3}}\right)\Bigg|_{x,p_x,z,p_z=0}=-\frac{1}{p_\theta},$$

$$p'_{r_{z^2}}=-\frac{1}{2}\frac{1}{p_\theta}p_z^2+\frac{1}{2}r\left(\vec{\nabla}_\perp^2 B\right)z^2+\frac{\partial B}{\partial\theta}\frac{1}{p_\theta}zp_z, \tag{7.19}$$

So p'_{r_2} which is second order terms in horizontal momentum is written as

$$p'_{r_2}=-\frac{1}{2}\frac{P^2}{p_\theta}p_x^2-\frac{1}{2}\left(2\frac{\partial B}{\partial r}-r\frac{\partial^2 B}{\partial r^2}\right)x^2-\frac{1}{2}\frac{1}{p_\theta}p_z^2+\frac{1}{2}r\left(\vec{\nabla}_\perp^2 B\right)z^2+\frac{\partial B}{\partial\theta}\frac{1}{p_\theta}zp_z. \tag{7.20}$$

z'_2 is derived such that

$$\mathrm{z}'=\frac{(r+x)\mathrm{p_z}}{\sqrt{P^2-(p_r+p_x)^2-p_z^2}}, \tag{7.21}$$

$$\begin{aligned}\mathrm{z}'_2=&\frac{1}{2}\frac{\partial^2}{\partial x^2}\left(\frac{dz}{d\theta}\right)\Big|_{x,p_x,z,p_z=0}x^2+\frac{\partial^2}{\partial p_x\partial x}\left(\frac{dz}{d\theta}\right)\Big|_{x,p_x,z,p_z=0}xp_x+\frac{1}{2}\frac{\partial^2}{\partial p_x^2}\left(\frac{dz}{d\theta}\right)\Big|_{x,p_x,z,p_z=0}p_x^2\\&+\frac{1}{2}\frac{\partial^2}{\partial z^2}\left(\frac{dz}{d\theta}\right)\Big|_{x,p_x,z,p_z=0}z^2+\frac{\partial^2}{\partial p_z\partial z}\left(\frac{dz}{d\theta}\right)\Big|_{x,p_x,z,p_z=0}zp_z+\frac{1}{2}\frac{\partial^2}{\partial p_z^2}\left(\frac{dz}{d\theta}\right)\Big|_{x,p_x,z,p_z=0}p_z^2\\&+\frac{\partial^2}{\partial z\partial x}\left(\frac{dz}{d\theta}\right)\Big|_{x,p_x,z,p_z=0}xz+\frac{\partial^2}{\partial p_z\partial x}\left(\frac{dz}{d\theta}\right)\Big|_{x,p_x,z,p_z=0}xp_z+\frac{\partial^2}{\partial z\partial p_x}\left(\frac{dz}{d\theta}\right)\Big|_{x,p_x,z,p_z=0}zp_x\\&+\frac{\partial^2}{\partial p_z\partial p_x}\left(\frac{dz}{d\theta}\right)\Big|_{x,p_x,z,p_z=0}p_xp_z,\end{aligned} \tag{7.22}$$

$$\begin{aligned}\frac{\partial^2}{\partial x^2}\left(\frac{(r+x)\mathrm{p_z}}{\sqrt{P^2-(p_r+p_x)^2-p_z^2}}\right)\Bigg|_{x,p_x,z,p_z=0}&=0\,,\\\frac{\partial^2}{\partial p_x\partial x}\left(\frac{(r+x)\mathrm{p_z}}{\sqrt{P^2-(p_r+p_x)^2-p_z^2}}\right)\Bigg|_{x,p_x,z,p_z=0}&=0,\end{aligned} \tag{7.23}$$

$$\frac{\partial^2}{\partial p_x^2}\left(\frac{(r+x)\mathrm{p_z}}{\sqrt{P^2-(p_r+p_x)^2-p_z^2}}\right)\Bigg|_{x,p_x,z,p_z=0} = 0,$$

$$\frac{\partial^2}{\partial p_z \partial x}\left(\frac{(r+x)\mathrm{p_z}}{\sqrt{P^2-(p_r+p_x)^2-p_z^2}}\right)\Bigg|_{x,p_x,z,p_z=0} = \frac{\partial}{\partial p_z}\left(\frac{\mathrm{p_z}}{\sqrt{P^2-(p_r+p_x)^2-p_z^2}}\right)\Bigg|_{x,p_x,z,p_z=0}$$

$$= \left(\frac{1}{\sqrt{P^2-(p_r+p_x)^2-p_z^2}} + \frac{\mathrm{p_z^2}}{\sqrt{\left(P^2-(p_r+p_x)^2-p_z^2\right)^3}}\right)\Bigg|_{x,p_x,z,p_z=0} = \frac{1}{p_\theta},$$

$$\frac{\partial^2}{\partial p_z \partial p_x}\left(\frac{(r+x)\mathrm{p_z}}{\sqrt{P^2-(p_r+p_x)^2-p_z^2}}\right)\Bigg|_{x,p_x,z,p_z=0} = \frac{\partial}{\partial p_z}\left(\frac{(r+x)p_z(p_r+p_x)}{\sqrt{\left(P^2-(p_r+p_x)^2-p_z^2\right)^3}}\right)\Bigg|_{x,p_x,z,p_z=0}$$

$$= \left(\frac{(r+x)(p_r+p_x)}{\sqrt{\left(P^2-(p_r+p_x)^2-p_z^2\right)^3}} + \frac{3(r+x)(p_r+p_x)p_z^2}{\sqrt{\left(P^2-(p_r+p_x)^2-p_z^2\right)^3}}\right)\Bigg|_{x,p_x,z,p_z=0} = \frac{rp_r}{p_\theta^3},$$

$$\frac{\partial^2}{\partial p_z^2}\left(\frac{(r+x)\mathrm{p_z}}{\sqrt{P^2-(p_r+p_x)^2-p_z^2}}\right)\Bigg|_{x,p_x,z,p_z=0} = \frac{\partial}{\partial p_z}\left(\frac{(r+x)}{\sqrt{P^2-(p_r+p_x)^2-p_z^2}} + \frac{(r+x)p_z^2}{\sqrt{\left(P^2-(p_r+p_x)^2-p_z^2\right)^3}}\right)\Bigg|_{x,p_x,z,p_z=0}$$

$$= \left(\frac{(r+x)p_z}{\sqrt{\left(P^2-(p_r+p_x)^2-p_z^2\right)^3}} + \frac{2(r+x)p_z}{\sqrt{\left(P^2-(p_r+p_x)^2-p_z^2\right)^3}} + \frac{6(r+x)p_z^3}{\sqrt{\left(P^2-(p_r+p_x)^2-p_z^2\right)^3}}\right)\Bigg|_{x,p_x,z,p_z=0} = 0,$$

$$\mathrm{z_2'} = \frac{1}{p_\theta}p_z x + \frac{rp_r}{p_\theta^3}p_x p_z = p_z\left(\frac{x}{p_\theta} + \frac{rp_r}{p_\theta^3}p_x\right). \tag{7.24}$$

p'_{z_2} is derived such that

$$p_z' = (r+x)z\frac{\partial B}{\partial r} - \frac{z(p_r+p_x)}{\sqrt{P^2-(p_r+p_x)^2-p_z^2}}\frac{\partial B}{\partial \theta}. \tag{7.25}$$

$$\begin{aligned} p'_{z_2} = {} & \frac{1}{2}\frac{\partial^2}{\partial x^2}\left(\frac{dp_z}{d\theta}\right)\Big|_{x,p_x,z,p_z=0} x^2 + \frac{\partial^2}{\partial p_x \partial x}\left(\frac{dp_z}{d\theta}\right)\Big|_{x,p_x,z,p_z=0} x p_x + \frac{1}{2}\frac{\partial^2}{\partial p_x^2}\left(\frac{dp_z}{d\theta}\right)\Big|_{x,p_x,z,p_z=0} p_x^2 \\ & + \frac{1}{2}\frac{\partial^2}{\partial z^2}\left(\frac{dp_z}{d\theta}\right)\Big|_{x,p_x,z,p_z=0} z^2 + \frac{\partial^2}{\partial p_z \partial z}\left(\frac{dp_z}{d\theta}\right)\Big|_{x,p_x,z,p_z=0} z p_z + \frac{1}{2}\frac{\partial^2}{\partial p_z^2}\left(\frac{dp_z}{d\theta}\right)\Big|_{x,p_x,z,p_z=0} p_z^2 \\ & + \frac{\partial^2}{\partial z \partial x}\left(\frac{dp_z}{d\theta}\right)\Big|_{x,p_x,z,p_z=0} xz + \frac{\partial^2}{\partial p_z \partial x}\left(\frac{dp_z}{d\theta}\right)\Big|_{x,p_x,z,p_z=0} x p_z + \frac{\partial^2}{\partial z \partial p_x}\left(\frac{dp_z}{d\theta}\right)\Big|_{x,p_x,z,p_z=0} z p_x \\ & + \frac{\partial^2}{\partial p_z \partial p_x}\left(\frac{dp_z}{d\theta}\right)\Big|_{x,p_x,z,p_z=0} p_x p_z, \end{aligned} \tag{7.26}$$

$$\frac{\partial^2}{\partial x^2}\left((r+x)z\frac{\partial B}{\partial r}-\frac{z(p_r+p_x)}{\sqrt{P^2-(p_r+p_x)^2-p_z^2}}\frac{\partial B}{\partial \theta}\right)\Bigg|_{x,p_x,z,p_z=0}=0,$$

$$\frac{\partial^2}{\partial p_x \partial x}\left((r+x)z\frac{\partial B}{\partial r}-\frac{z(p_r+p_x)}{\sqrt{P^2-(p_r+p_x)^2-p_z^2}}\frac{\partial B}{\partial \theta}\right)\Bigg|_{x,p_x,z,p_z=0}=0,$$

$$\frac{\partial^2}{\partial p_x^2}\left((r+x)z\frac{\partial B}{\partial r}-\frac{z(p_r+p_x)}{\sqrt{P^2-(p_r+p_x)^2-p_z^2}}\frac{\partial B}{\partial \theta}\right)\Bigg|_{x,p_x,z,p_z=0}=0,$$

$$\frac{\partial^2}{\partial z^2}\left((r+x)z\frac{\partial B}{\partial r}-\frac{z(p_r+p_x)}{\sqrt{P^2-(p_r+p_x)^2-p_z^2}}\frac{\partial B}{\partial \theta}\right)\Bigg|_{x,p_x,z,p_z=0}=0,$$

$$\frac{\partial^2}{\partial p_z \partial z}\left((r+x)z\frac{\partial B}{\partial r}-\frac{z(p_r+p_x)}{\sqrt{P^2-(p_r+p_x)^2-p_z^2}}\frac{\partial B}{\partial \theta}\right)\Bigg|_{x,p_x,z,p_z=0}=0,$$

$$\frac{\partial^2}{\partial p_z^2}\left((r+x)z\frac{\partial B}{\partial r}-\frac{z(p_r+p_x)}{\sqrt{P^2-(p_r+p_x)^2-p_z^2}}\frac{\partial B}{\partial \theta}\right)\Bigg|_{x,p_x,z,p_z=0}=0, \tag{7.27}$$

$$\frac{\partial^2}{\partial x \partial z}\left((r+x)z\frac{\partial B}{\partial r}-\frac{z(p_r+p_x)}{\sqrt{P^2-(p_r+p_x)^2-p_z^2}}\frac{\partial B}{\partial \theta}\right)\Bigg|_{x,p_x,z,p_z=0}$$

$$=\frac{\partial}{\partial x}\left((r+x)\frac{\partial B}{\partial r}-\frac{(p_r+p_x)}{\sqrt{P^2-(p_r+p_x)^2-p_z^2}}\frac{\partial B}{\partial \theta}\right)\Bigg|_{x,p_x,z,p_z=0}$$

$$=\left(\frac{\partial B}{\partial r}+(r+x)\frac{\partial^2 B}{\partial x \partial r}-\frac{(p_r+p_x)}{\sqrt{P^2-(p_r+p_x)^2-p_z^2}}\frac{\partial^2 B}{\partial x \partial \theta}\right)\Bigg|_{x,p_x,z,p_z=0}=\frac{\partial B}{\partial r}+r\frac{\partial^2 B}{\partial r^2}-\frac{p_r}{p_\theta}\frac{\partial^2 B}{\partial r \partial \theta},$$

$$\frac{\partial^2}{\partial p_x \partial z}\left((r+x)z\frac{\partial B}{\partial r}-\frac{z(p_r+p_x)}{\sqrt{P^2-(p_r+p_x)^2-p_z^2}}\frac{\partial B}{\partial \theta}\right)\Bigg|_{x,p_x,z,p_z=0}$$

$$=\frac{\partial}{\partial p_x}\left((r+x)\frac{\partial B}{\partial r}-\frac{(p_r+p_x)}{\sqrt{P^2-(p_r+p_x)^2-p_z^2}}\frac{\partial B}{\partial \theta}\right)\Bigg|_{x,p_x,z,p_z=0}$$

$$=\left(-\frac{(p_r+p_x)^2}{\sqrt{\left(P^2-(p_r+p_x)^2-p_z^2\right)^3}}\frac{\partial B}{\partial \theta}\right)\Bigg|_{x,p_x,z,p_z=0}=-\frac{p_r^2}{p_\theta^3}\frac{\partial B}{\partial \theta},$$

$$p'_{z_2}=\left(\frac{\partial B}{\partial r}+r\frac{\partial^2 B}{\partial r^2}-\frac{p_r}{p_\theta}\frac{\partial^2 B}{\partial r \partial \theta}\right)xz-\frac{p_r^2}{p_\theta^3}\frac{\partial B}{\partial \theta}zp_x. \tag{7.28}$$

Therefore, we can get the second order terms (r'_2, p'_{r_2}, z'_2 and p'_{z_2}) is given by

$$
\begin{aligned}
r_2' &= \frac{P^2}{p_\theta^3} x p_x + \frac{3}{2}\frac{r p_r P^2}{p_\theta^5} p_x^2 + \frac{1}{2}\frac{r p_r}{p_\theta^3} p_z^2. \\
p_{r_2}' &= -\frac{1}{2}\frac{P^2}{p_\theta} p_x^2 - \frac{1}{2}\left(2\frac{\partial B}{\partial r} - r\frac{\partial^2 B}{\partial r^2}\right) x^2 - \frac{1}{2}\frac{1}{p_\theta} p_z^2 + \frac{1}{2} r \left(\vec{\nabla}_\perp^2 B\right) z^2 + \frac{\partial B}{\partial \theta}\frac{1}{p_\theta} z p_z, \\
z_2' &= \frac{1}{p_\theta} p_z x + \frac{r p_r}{p_\theta^3} p_x p_z = p_z\left(\frac{x}{p_\theta} + \frac{r p_r}{p_\theta^3} p_x\right), \\
p_{z_2}' &= \left(\frac{\partial B}{\partial r} + r\frac{\partial^2 B}{\partial r^2} - \frac{p_r}{p_\theta}\frac{\partial^2 B}{\partial r \partial \theta}\right) x z - \frac{p_r^2}{p_\theta^3}\frac{\partial B}{\partial \theta} z p_x.
\end{aligned}
\tag{7.29}
$$

7.3 Solving the non-linear equation of motion

To non-linear terms of Eqs. (4.1~4), we transform Eqs. (4.1~4) from non-linear equations to linear inhomogeneous equations. The beam's equations of motion to second order are written as

$$\begin{aligned}
\frac{dx}{d\theta} &= \frac{p_r}{p_\theta}x + \left(\frac{rP^2}{p_\theta^3}\right)p_x + \frac{P^2}{p_\theta^3}xp_x + \frac{3}{2}\frac{rp_rP^2}{p_\theta^5}p_x^2 + \frac{1}{2}\frac{rp_r}{p_\theta^3}p_z^2.\\
\frac{dp_x}{d\theta} &= -\frac{p_r}{p_\theta}p_x - \left(B + r\frac{\partial B}{\partial r}\right)x - \frac{1}{2}\frac{P^2}{p_\theta}p_x^2 - \frac{1}{2}\left(2\frac{\partial B}{\partial r} - r\frac{\partial^2 B}{\partial r^2}\right)x^2 - \frac{1}{2}\frac{1}{p_\theta}p_z^2 + \frac{1}{2}r\left(\vec{\nabla}_\perp^2 B\right)z^2 + \\
&\qquad \frac{\partial B}{\partial\theta}\frac{1}{p_\theta}zp_z, \qquad (7.30)\\
\frac{dz}{d\theta} &= \frac{r}{p_\theta}p_z + p_z\left(\frac{x}{p_\theta} + \frac{rp_r}{p_\theta^3}p_x\right),\\
\frac{dp_z}{d\theta} &= \left(r\frac{\partial B}{\partial r} - \frac{p_r}{p_\theta}\frac{\partial B}{\partial\theta}\right)z + \left(\frac{\partial B}{\partial r} + r\frac{\partial^2 B}{\partial r^2} - \frac{p_r}{p_\theta}\frac{\partial^2 B}{\partial r\partial\theta}\right)xz - \frac{p_r^2}{p_\theta^3}\frac{\partial B}{\partial\theta}zp_x.
\end{aligned}$$

Inhomogeneous equation [Eqs. (4.1~4)] are derive as

$$\begin{aligned}
\frac{dx}{d\theta} &= \frac{p_r}{p_\theta}x + \frac{rP^2}{p_\theta^3}p_x + f_x(\theta),\\
f_x(\theta) &= \frac{\mathrm{P}^2}{p_\theta^3}xp_x + \frac{3}{2}\frac{rp_rP^2}{P_\theta^5}p_x^2 + \frac{1}{2}\frac{rp_r}{p_\theta^3}p_z^2,\\
f_x(\theta) &= \frac{\mathrm{P}^2}{p_\theta^3}(X_{11}x_0 + X_{12}p_{x0})(X_{21}x_0 + X_{22}p_{x0}) + \frac{3}{2}\frac{rp_rP^2}{P_\theta^5}(X_{21}x_0 + X_{22}p_{x0})^2\\
&\quad + \frac{1}{2}\frac{rp_r}{p_\theta^3}(Z_{21}z_0 + Z_{22}p_{z0})^2 \qquad (7.31)\\
&= \left(\frac{\mathrm{P}^2}{p_\theta^3}X_{11}X_{21} + \frac{3}{2}\frac{rp_rP^2}{p_\theta^5}X_{21}^2\right)x_0^2 + \left(\frac{\mathrm{P}^2}{p_\theta^3}(X_{11}X_{22} + X_{12}X_{21}) + 3\frac{rp_rP^2}{P_\theta^5}X_{21}X_{22}\right)x_0p_{x0}\\
&+ \left(\frac{\mathrm{P}^2}{p_\theta^3}X_{12}X_{22} + \frac{3}{2}\frac{rp_rP^2}{P_\theta^5}X_{22}^2\right)p_{x0}^2 + \left(\frac{1}{2}\frac{rp_r}{p_\theta^3}Z_{21}^2\right)z_0^2 + \left(\frac{rp_r}{p_\theta^3}Z_{21}Z_{22}\right)z_0p_{z0} + \left(\frac{1}{2}\frac{rp_r}{p_\theta^3}Z_{22}^2\right)p_{zo}^2.\\
&= f_{11}x_0^2 + f_{12}x_0p_{x0} + f_{13}p_{x0}^2 + f_{14}z_0^2 + f_{15}z_0p_{z0} + f_{16}p_{zo}^2.
\end{aligned}$$

$$\frac{dp_x}{d\theta} = -\frac{p_r}{p_\theta}p_x - \left(B_z + r\frac{\partial B_z}{\partial r}\right)x + f_{p_x}(\theta),$$

$$\begin{aligned}
f_{p_x}(\theta) &= -\frac{1}{2}\frac{P^2}{p_\theta^3}p_x^2 - \frac{1}{2}(2\frac{\partial B_0}{\partial r} + r\frac{\partial^2 B_0}{\partial x^2})x^2 - \frac{1}{2}\frac{1}{p_\theta}p_z^2 + \frac{1}{2}r\left(\vec{\nabla}^2 B_z\right)z^2 + \frac{\partial B_0}{\partial\theta}\frac{1}{p_\theta}zp_z,\\
f_{p_x}(\theta) &= -\frac{1}{2}\left(2\frac{\partial B_0}{\partial r} + r\frac{\partial^2 B_0}{\partial x^2}\right)(X_{11}x_0 + X_{12}p_{x0})^2 - \frac{1}{2}\frac{P^2}{p_\theta^3}(X_{21}x_0 + X_{22}p_{x0})^2 +\\
&\quad \frac{1}{2}r\left(\vec{\nabla}^2 B_z\right)(Z_{11}(\theta)z_0 + Z_{12}(\theta)p_{z0})^2 + \frac{\partial B_0}{\partial\theta}\frac{1}{p_\theta}(Z_{11}(\theta)z_0 + Z_{12}(\theta)p_{z0})(Z_{21}(\theta)z_0 +\\
&\quad Z_{22}(\theta)p_{z0}) - \frac{1}{2}\frac{1}{p_\theta}(Z_{21}(\theta)z_0 + Z_{22}(\theta)p_{z0})^2 \qquad (7.32)\\
&\quad = \left(-\frac{1}{2}\left(2\frac{\partial B_0}{\partial r} + r\frac{\partial^2 B_0}{\partial x^2}\right)X_{11}^2 - \frac{1}{2}\frac{P^2}{p_\theta^3}X_{21}^2\right)x_0^2 + 2\times\left(-\frac{1}{2}\left(2\frac{\partial B_0}{\partial r} + r\frac{\partial^2 B_0}{\partial x^2}\right)X_{11}X_{12} -\right.\\
&\quad \left.\frac{1}{2}\frac{P^2}{p_\theta^3}X_{21}X_{22}\right)x_0p_{x0} + \left(-\frac{1}{2}\left(2\frac{\partial B_0}{\partial r} + r\frac{\partial^2 B_0}{\partial x^2}\right)X_{12}^2 - \frac{1}{2}\frac{P^2}{p_\theta^3}X_{22}^2\right)p_{x0}^2 + \left(\frac{1}{2}r\left(\vec{\nabla}^2 B_z\right)Z_{11}^2 +\right.\\
&\quad \left.\frac{\partial B_0}{\partial\theta}\frac{1}{p_\theta}Z_{11}Z_{21} - \frac{1}{2}\frac{1}{p_\theta}Z_{21}^2\right)z_0^2 + (2\times\frac{1}{2}r\left(\vec{\nabla}^2 B_z\right)Z_{11}Z_{12} + \frac{\partial B_0}{\partial\theta}\frac{1}{p_\theta}(Z_{11}Z_{22} +\\
&\quad Z_{12}Z_{21})z_0p_{z0} + (\frac{1}{2}r\left(\vec{\nabla}^2 B_z\right)Z_{12}^2 + \frac{\partial B_0}{\partial\theta}\frac{1}{p_\theta}Z_{12}Z_{22} - \frac{1}{2}\frac{1}{p_\theta}Z_{22}^2)p_{z0}^2\\
&\quad = f_{21}x_0^2 + f_{22}x_0p_{x0} + f_{23}p_{x0}^2 + f_{24}z_0^2 + f_{25}z_0p_{z0} + f_{26}p_{zo}^2.
\end{aligned}$$

$$\frac{dp_z}{d\theta} = \left(r\frac{\partial B}{\partial r} - \frac{p_r}{p_\theta}\frac{\partial B}{\partial\theta}\right)z + f_{p_z}(\theta),$$

$$\begin{aligned}
f_{p_z}(\theta) &= \frac{1}{p_\theta}xp_z + \frac{rp_r}{p_\theta^3}p_xp_z,\\
f_z(\theta) &= \frac{1}{p_\theta}(X_{11}x_0 + X_{12}p_{x0})(Z_{21}z_0 + Z_{22}p_{z0}) + \frac{rp_r}{p_\theta^3}(X_{21}x_0 + X_{22}p_{x0})(Z_{21}z_0 +\\
&\quad Z_{22}p_{z0}) \qquad (7.33)\\
&\quad = (\frac{1}{p_\theta}X_{11}Z_{21} + \frac{rp_r}{p_\theta^3}X_{21}Z_{21})x_0z_0 + \left(\frac{1}{p_\theta}X_{11}Z_{22} + \frac{rp_r}{p_\theta^3}X_{21}Z_{22}\right)x_0p_{z_0} +\\
&\quad \left(\frac{1}{p_\theta}X_{12}Z_{21} + \frac{rp_r}{p_\theta^3}X_{22}Z_{21}\right)p_{x_0}z_0 + \left(\frac{1}{p_\theta}X_{12}Z_{22} + \frac{rp_r}{p_\theta^3}X_{22}Z_{22}\right)p_{x_0}p_{z_0},\\
&= g_{11}x_0z_0 + g_{12}x_0p_{z_0} + g_{13}p_{x_0}z_0 + g_{14}p_{x_0}p_{z_0}.
\end{aligned}$$

$$\frac{dp_z}{d\theta} = \frac{r}{p_\theta} p_z + f_z(\theta),$$

$$\begin{aligned}
f_z(\theta) &= \left(\frac{\partial B}{\partial r} + r\frac{\partial^2 B}{\partial r^2} - \frac{p_r}{p_\theta}\frac{\partial^2 B}{\partial r \partial \theta}\right) xz - \frac{p_r^2}{p_\theta^3}\frac{\partial B}{\partial \theta} z p_x, \\
f_z(\theta) &= \left(\frac{\partial B_0}{\partial r} + r\frac{\partial^2 B_0}{\partial r^2} - \frac{p_r}{p_\theta}\frac{\partial^2 B_0}{\partial r \partial \theta}\right)(X_{11}x_0 + X_{12}p_{x0})(Z_{11}z_0 + Z_{12}p_{z0}) - \frac{p_r^2}{p_\theta^3}\frac{\partial B_0}{\partial \theta}(Z_{11}z_0 + \\
&\quad Z_{12}p_{z0})(X_{21}x_0 + X_{22}p_{x0}) \\
&= \left(\left(\frac{\partial B_0}{\partial r} + r\frac{\partial^2 B_0}{\partial r^2} - \frac{p_r}{p_\theta}\frac{\partial^2 B_0}{\partial r \partial \theta}\right) X_{11}Z_{11} - \frac{p_r^2}{p_\theta^3}\frac{\partial B_0}{\partial \theta} X_{21}Z_{11}\right) x_0 z_0 + \left(\left(\frac{\partial B_0}{\partial r} + r\frac{\partial^2 B_0}{\partial r^2} - \right.\right. \\
&\quad \left.\left.\frac{p_r}{p_\theta}\frac{\partial^2 B_0}{\partial r \partial \theta}\right) X_{11}Z_{12} - \frac{p_r^2}{p_\theta^3}\frac{\partial B_0}{\partial \theta} X_{21}Z_{11}\right) x_0 p_{z0} + + \left(\left(\frac{\partial B_0}{\partial r} + r\frac{\partial^2 B_0}{\partial r^2} - \frac{p_r}{p_\theta}\frac{\partial^2 B_0}{\partial r \partial \theta}\right) X_{12}Z_{11} - \right. \\
&\quad \left.\frac{p_r^2}{p_\theta^3}\frac{\partial B_0}{\partial \theta} X_{22}Z_{11}\right) p_{x0} z_0 + \left(\left(\frac{\partial B_0}{\partial r} + r\frac{\partial^2 B_0}{\partial r^2} - \frac{p_r}{p_\theta}\frac{\partial^2 B_0}{\partial r \partial \theta}\right) X_{12}Z_{22} - \right. \\
&\quad \left.\frac{p_r^2}{p_\theta^3}\frac{\partial B_0}{\partial \theta} X_{22}Z_{12}\right) p_{x0} p_{z0}, \\
&= g_{21}x_0 z_0 + g_{22}x_0 p_{z0} + g_{23}p_{x_0} z_0 + g_{24}p_{x_0}p_{z_0}.
\end{aligned} \tag{7.34}$$

Therefore, the non-linear equations of motion are written as

$$\begin{aligned}
\frac{dx}{d\theta} &= \frac{p_r}{p_\theta}x + \frac{rP^2}{p_\theta^3}p_x + f_{11}x_0^2 + f_{12}x_0 p_{x0} + f_{13}p_{x0}^2 + f_{14}z_0^2 + f_{15}z_0 p_{z0} + f_{16}p_{zo}^2, \\
\frac{dp_x}{d\theta} &= -\frac{p_r}{p_\theta}p_x - \left(B_z + r\frac{\partial B_z}{\partial r}\right)x + f_{21}x_0^2 + f_{22}x_0 p_{x0} + f_{23}p_{x0}^2 + f_{24}z_0^2 + f_{25}z_0 p_{z0} + \\
&\quad f_{26}p_{zo}^2,
\end{aligned} \tag{7.35}$$

$$\begin{aligned}
\frac{dp_z}{d\theta} &= \left(r\frac{\partial B}{\partial r} - \frac{p_r}{p_\theta}\frac{\partial B}{\partial \theta}\right) + g_{11}x_0 z_0 + g_{12}x_0 p_{z_0} + g_{13}p_{x_0} z_0 + g_{14}p_{x_0}p_{z_0}, \\
\frac{dp_z}{d\theta} &= \frac{r}{p_\theta}p_z + g_{21}x_0 z_0 + g_{22}x_0 p_{z0} + g_{23}p_{x_0} z_0 + g_{24}p_{x_0}p_{z_0}.
\end{aligned} \tag{7.36}$$

The Eq. (7.35) is represented matrix form such that

$$\begin{aligned}
\begin{pmatrix} \frac{dx}{d\theta} \\ \frac{dp_x}{d\theta} \end{pmatrix} &= \begin{pmatrix} \frac{p_r}{p_\theta} & \frac{rP^2}{p_\theta^3} \\ -\frac{p_r}{p_\theta}p_x & -\left(B_z + r\frac{\partial B_z}{\partial r}\right) \end{pmatrix} \begin{pmatrix} x \\ p_x \end{pmatrix} + \begin{pmatrix} f_{11} \\ f_{21} \end{pmatrix} x_0^2 + \begin{pmatrix} f_{12} \\ f_{22} \end{pmatrix} x_0 p_{x_0} + \begin{pmatrix} f_{13} \\ f_{23} \end{pmatrix} p_{x_0}^2 + \\
&\quad \begin{pmatrix} f_{14} \\ f_{24} \end{pmatrix} z_0^2 + \begin{pmatrix} f_{15} \\ f_{25} \end{pmatrix} z_0 p_{z_0} + \begin{pmatrix} f_{16} \\ f_{26} \end{pmatrix} p_{z_0}^2,
\end{aligned} \tag{7.37}$$

Considering only one term of the non-linear part, the equation is given by

$$\begin{pmatrix} \frac{dx}{d\theta} \\ \frac{dp_x}{d\theta} \end{pmatrix} = K(\theta)\begin{pmatrix} x \\ p_x \end{pmatrix} + \begin{pmatrix} f_{11} \\ f_{21} \end{pmatrix} x_0^2. \tag{7.38}$$

And the solution of the matrix [Eq. (7.38)] can be written as

$$\begin{pmatrix} x(\theta) \\ p_x(\theta) \end{pmatrix} = \begin{pmatrix} X_{11}(\theta) & X_{12}(\theta) \\ X_{21}(\theta) & X_{22}(\theta) \end{pmatrix}\begin{pmatrix} x_0 \\ p_{x_0} \end{pmatrix} + \left(\int_0^\theta X(\theta)X^{-1}(\theta')\begin{pmatrix} f_{11}(\theta') \\ f_{21}(\theta') \end{pmatrix} d\theta'\right) x_0^2. \tag{7.39}$$

In Eq. (7.39), the first term is called homogeneous solution, and second term is called inhomogeneous solution. The inhomogeneous solution can be proven as

$$\begin{aligned} &\begin{pmatrix} x(\theta) \\ p_x(\theta) \end{pmatrix} = \left(\int_0^\theta X(\theta)X^{-1}(\theta')\begin{pmatrix} f_{11}(\theta') \\ f_{21}(\theta') \end{pmatrix} d\theta'\right) x_0^2, \\ &\begin{pmatrix} \frac{dx(\theta)}{d\theta} \\ \frac{dp_x(\theta)}{d\theta} \end{pmatrix} = \frac{d}{d\theta}\left(\int_0^\theta X(\theta)X^{-1}(\theta')\begin{pmatrix} f_{11}(\theta') \\ f_{21}(\theta') \end{pmatrix} d\theta'\right) x_0^2 \\ &= X(\theta)X^{-1}(\theta)\begin{pmatrix} f_{11}(\theta') \\ f_{21}(\theta') \end{pmatrix} x_0^2 + \left(\int_0^\theta \frac{dX(\theta)}{d\theta}X^{-1}(\theta')\begin{pmatrix} f_{11}(\theta') \\ f_{21}(\theta') \end{pmatrix} d\theta'\right) x_0^2 \\ &= \begin{pmatrix} f_{11}(\theta') \\ f_{21}(\theta') \end{pmatrix} x_0^2 + K(\theta)\left(\int_0^\theta X(\theta)X^{-1}(\theta')\begin{pmatrix} f_{11}(\theta') \\ f_{21}(\theta') \end{pmatrix} d\theta'\right) x_0^2 \\ &= K(\theta)\begin{pmatrix} x(\theta) \\ p_x(\theta) \end{pmatrix} + \begin{pmatrix} f_{11}(\theta') \\ f_{21}(\theta') \end{pmatrix} x_0^2. \end{aligned} \tag{7.40}$$

Also, we define A_n and B_n as

$$\begin{aligned} A_n &= \begin{pmatrix} a_{1n} \\ a_{2n} \end{pmatrix} = \left(\int_0^\theta X(\theta)X^{-1}(\theta')\begin{pmatrix} f_{1n}(\theta') \\ f_{2n}(\theta') \end{pmatrix} d\theta'\right), \\ B_n &= \begin{pmatrix} b_{1n} \\ b_{2n} \end{pmatrix} = \left(\int_0^\theta Z(\theta)Z^{-1}(\theta')\begin{pmatrix} g_{1n}(\theta') \\ g_{2n}(\theta') \end{pmatrix} d\theta'\right). \end{aligned} \tag{7.41}$$

So, the complete solution of second order equation of motion can be written as

$$\begin{aligned} \begin{pmatrix} x(\theta) \\ p_x(\theta) \end{pmatrix} &= X(\theta)\begin{pmatrix} x_0 \\ p_{x_0} \end{pmatrix} + A_1 x_0^2 + A_2 x_0 p_{x_0} + A_3 p_{x_0}^2 + A_4 z_0^2 + A_5 z_0 p_{z_0} + A_6 p_{z_0}^2, \\ \begin{pmatrix} z(\theta) \\ p_z(\theta) \end{pmatrix} &= Z(\theta)\begin{pmatrix} z_0 \\ p_{z_0} \end{pmatrix} + B_1 x_0 z_0 + B_2 x_0 p_{z_0} + B_3 p_{x_0} z_0 + B_4 p_{x_0} p_{z_0}. \end{aligned} \tag{7.42}$$

7.4 Second order code description

In this section, we wrote a summary script that is essential for running the code.

The script of code and field data can be downloaded from Index of /code-scripts/Cyclotron.

```
% Start program
% Define parameters.
rms=0.05; %rms of initial gaussian beam distribution x, z
rmspx=0.0001; %rms of Px
rmspz=0.0001; %rms of Pz
plot_eo=123; % To see orbit you want
order=1; % Calculation order 1=first order, 2=second order, 3=Full equation
Npar=5000; % Particle number
Neo=123; % Maximum turns


% Ark, Brk and Crk are integration coefficients (Runge-Kutta-Gill).
% Define parameters and dimension.
Tpi = 6.28318530718D+0;
Ark = [0.5 ; 0.292893219 ; 1.707106781 ; 0.1666666667];
Brk = [2.0 ; 1.0 ; 1.0 ; 2.0];
Crk = [-0.5 ; -0.292893219 ; -1.707106781 ; -0.5];

% Open output file
Output_data1 = fopen('eo.dat','w');
Output_data2 = fopen('eodat.dat','w');


% Read field parameters and set integration step.
field = fopen('field.dat','r');
B=fscanf(field,'%f',[360,36]);
%% Plot magnetic field.
figure;
contour(XX,YY,B);
title('Magnetic field conture plot')
xlabel('x [cm]')
ylabel('y [cm]')


% Get input data.
% Number of sector(1 to 4),Number of theta values in the field,
% Number of radius values in the field,Radius of first value in
field(inches)
% Radius increment(inches),Rest mass of particle(MeV,931.5 is standard)
% Q/E Charge of particle(),B constant from RF Field unit(kilogauss)
Nsc = 1;      % Number of sector (1 to 4)
Nth = 360;    % Number of theta values in the field
Nr = 36;      % Number of radius values in the field
R0 = 1.0;     % Radius of first value in field (inches)
Dr = 0.5;     % Radius increment (inches)
E0 = 939.3016; % Rest mass of particle (MeV, 931.5 is standard)
Chg = 1.0;    % Charge of particle
Bcon = 12.679; % B constant


% Read fields; Note first value(theta=0)stored at I=2.
% Add extra points for taking theta derivatives.
for J = 2:Nr % Nr = 37, (real Nr) + 1
```

```
    for I = 2:Nt1 % Nt1 = 361
    C(I,J)=B(I-1,J-1);  %% check is needed...
                    % C(2,2)=B(1,1), C(Nth+1,Nr+1)=B(Nth,Nr)
    end
    C(1,J)=B(Nt1-1,J-1);  % C(1,r+1)=B(Nth,r)
    C(Nt1+1,J)=B(2,J-1);  % C(Nth+2,r+1)=B(2,r)
end
B=C;

Nt1 = Nth + 2;

% Add extra points for R-interpolation at center.
for I = 1:Nt1
    B(I,1)=3*(B(I,2)-B(I,3))+B(I,4);  %(B1-B4)/(1-4)=(B2-B3)/(2-3)
end

% Divide field by field unit bcon.
for J = 1:Nr
    for I = 1:Nt1
       B(I,J)=B(I,J)/Bcon; %B => B/(m_0*w_0)/e)
    end
end


% Define some input values
% 0: n(stp)
% 1: E,dE,nE
% 2: r,Pr
% 3: Tht0,dtht,Ntht
% 4: Ip,Ii (start orbit run)
% 5: Exit program
P(1,1)=180;
P(2,1)=0;
P(3,1)=0;
P(1,2)=0.31; %initial kinetic energy
%P(1,2)=1.0;
P(2,2)=0.1; %Energy Step
%P(3,2)=300;
P(3,2)=Neo; % Number of Equilibrium Orbit
P(1,3)=4; % 1st initial r
%P(1,3)=2.5; % 1st initial r
P(2,3)=-0.03; % 1st initial Pr

% Setup constants for orbit calculation
Acon=E0*1000./(Chg*Bcon*299.7925*2.54); %Chg=charge=1, 1inch=2.54cm
Anu0=299.7925/(6.283185*0.0254*Acon); % Anu0=c/(2pi*Acon)
E=P(1,2); % user-supplied initial kinetic energy

% Set initial values for table.
Rg = P(1,3); %user-supplied initial r
Prg = P(2,3); %user-supplied initial Pr

% Loop for finding orbit parameters at each energy.
Xf1=zeros(2,Npar,Neo);
Zf1=zeros(2,Npar,Neo);

for ieo = 1:Neo
```

```
    Iter = 20; % Number of iteration steps.
    KK = 0;
    Psq = E/E0; % E = "KINETIC" Energy
    Gam = 1.D+0 + Psq; % Gam = Relativistic factor
    Psq = Psq*(2.D+0 + Psq)*Acon^2; % P^2 = (Gam^2 - 1)*Acon^2 = (Gam^2 -
1)*(mc/(mw_0))^2
    Ntry = 0;

    YY=zeros(4,Npar);
    if(successive==0 || ieo==1)
       xi      = rms*randn(1,Npar);   %Gaussian distribution for x
       xpi     = rmspx*randn(1,Npar); % for Px
       zi      = rms*randn(1,Npar);   % for z
       zpi     = rmspz*randn(1,Npar); % for Pz
       YY(1,:) = rms*randn(1,Npar);   % (Full equation) Gaussian
distrubution for x
       YY(2,:) = rmspx*randn(1,Npar); % (Full equation) for Px
       YY(3,:) = rms*randn(1,Npar);   % (Full equation) for z
       YY(4,:) = rmspz*randn(1,Npar); % (Full equation) for Pz

     %% Calcultate RMS emittance for initail value
       Axi=sum(xi.^2)/5000;          % <x^2>
       Apxi=sum(xpi.^2)/5000;        % <x'^2>

       xxi=xi';
       Axpx2i=xpi*xxi/5000;          % <xx'>
       Axpx2i=Axpx2i^2;             % <xx'>^2
       Ermsxi=sqrt(Axi*Apxi-Axpx2i)   % RMS emittance root(<x^2><x'^2> -
<xx'>^2)


       Azi=sum(zi.^2)/5000;          % <z^2>
       Apzi=sum(zpi.^2)/5000;        % <z'^2>

       zzi=zi';
       Azpz2i=zpi*zzi/5000;          % <zz'>
       Azpz2i=Azpz2i^2;             % <zz'>^2
       Ermszi=sqrt(Azi*Apzi-Azpz2i)   % RMS emittance root(<z^2><z'^2> -
<zz'>^2)

while (KK < Iter) & (Ntry ~= -1)
       KK = KK + 1;

       %
       % Set initial values of integration variables.
       % (Come back here after each Eo).
       Y(1) = Rg;    % Initial position. = 4
       Y(2) = Prg;   % Initial monentum. = -0.03
       Y(3) = 1.D+0; % x1
       Y(4) = 0.D+0; % px1
       Y(5) = 0.D+0; % x2
       Y(6) = 1.D+0; % px2
       Y(7) = 0.D+0;
       Y(8) = 1.D+0; % z1
       Y(9) = 0.D+0; % pz1
       Y(10)= 0.D+0; % z2
       Y(11)= 1.D+0; % Pz2
```

```
        Y(12)= 0.D+0;

        % Time eq.
        Y(13)=0.D+0;
        Nxcros=0;
        Nzcros=0;
        Bavet=0;

        for I =1:13
          Q(I)=0.0;
        end
        if (order==3)
          QQ=zeros(4,Npar);
        end
        if (order==2)
          INT_GX=zeros(2,6);
          INT_GZ=zeros(2,4);
        end
        if (Ntry==-2 && order==3)
           for k=1:Npar
             YY(1,k)=YY(1,k)+Y(1);
             YY(2,k)=YY(2,k)+Y(2);
           end
        end
        %
        % Loop for finding orbit at each theta.
        %

        for Istp = 1:Nstp
            Istp=Istp-1;

        %
        % Begin integration step.
        %
            Rst(Istp+1)=Y(1); % Y(1) = Initial position. = 4
            Prst(Istp+1)=Y(2); % Y(2) = Initial monentum. = -0.03
            Yp5=Y(5); % x2
            Yp10=Y(10); % px2

            if (Ntry ==-2)
               Cox(ieo,Istp+1)=Rst(Istp+1);
               Cop(ieo,Istp+1)=Prst(Istp+1);
               coxx(ieo,Istp+1)=2.54*Cox(ieo,Istp+1)*cos((Istp+1)*2*pi/180);

coyy(ieo,Istp+1)=2.54*Cox(ieo,Istp+1)*sin((Istp+1)*2*pi/180); %2.54 =
[inch]-> [cm]
               orbitx(ieo,Istp+1)=2.54*YY(1,1)*cos((Istp+1)*2*pi/180); % 1
particle orbit x
               orbity(ieo,Istp+1)=2.54*YY(1,1)*sin((Istp+1)*2*pi/180); % 1
particle orbit y
               orbitr(ieo,Istp+1)=YY(1,1);
            end

             if (Ntry==-2 && order==3 && cal_cen==1)
                for i = 1:Npar
                  Xftf(1,i,Istp+1)=YY(1,i)-Y(1);
                  Xftf(2,i,Istp+1)=YY(2,i)-Y(2);
                  Zftf(1,i,Istp+1)=YY(3,i);
```

```
        Zftf(2,i,Istp+1)=YY(4,i);
      end
    end

if (Ntry == -2)
     Time1= Y(13); % tau=w_0*t
  end
  %
  % Runge-Kutta iteration
  %
  for j = 1:4
     Ith=fix(j/2);   % lth = 0,1,1,2
     Ith=2*Istp+Ith+2; % lth = 2*step+lth+2

     while (Ith >= Nth+2)
        Ith=Ith-Nth;
     end

     Y1=Y(1);
     DR = 0.5; %%%%%%%%%%%%%%%
     Frb=(Y1-R0)/DR-1; % Y1=r R0=1 DR=0.5
     if (Frb < 0)
        fprintf('Error Off field in low postion\n');
        pause;
     end
     Nrb=fix(Frb); % Frb = 2*(r-1)-1 = 2r-3 ??
     Frb=Frb-Nrb;

     if ( Nrb+4-Nr > 0)
        fprintf('Error Off field in high postion\n');
        pause;
     end
     %
     % R-interpolation for B-field
     Frb2=Frb*Frb;
     Frb3=Frb*Frb2;
     Cof(1)=Frb2-0.5*(Frb3+Frb);
     Cof(2)=1.5*Frb3-2.5*Frb2+1.0;
     Cof(3)=2.0*Frb2-1.5*Frb3+0.5*Frb;
     Cof(4)=0.5*(Frb3-Frb2);
     Cof(5)=2.0*Frb-1.5*Frb2-0.5;
     Cof(6)=4.5*Frb2-5.0*Frb;
     Cof(7)=4.0*Frb-4.5*Frb2+0.5;
     Cof(8)=1.5*Frb2-Frb;
     Cof(9)=2.0-3.0*Frb;
     Cof(10)=9.0*Frb-5.0;
     Cof(11)=4.0-9.0*Frb;
     Cof(12)=3.0*Frb-1;

     Bint=0.0; % B
     Brint=0.0; % r*dB/dr
     Br=0.0; % dB/dr
     Brr=0.0; % d^2B/dr^2

     for i=1:4
        Bti=B(Ith,Nrb+i);
        Bint=Bint+Cof(i)*Bti;
        Brint=Brint+Cof(i+4)*Bti;
```

```
                Br=Br+Cof(i+4)*Bti;
                Brr=Brr+Cof(i+8)*Bti;
            end
            Brint=Brint*Y1/DR; %r*(dB/dr)
            Br = Br/DR;
            Br1(Ith) = Br;
            Brr=Brr/(DR*DR);
            Brr1(Ith) = Brr;
            % Theta derivative (Only when z equations used)
            if (Ntry == -2)
              Bt=0.0; % dB/dth
              Btt=0.0; % d^2B/dth^2
              Brt=0.0; % d^2B/drdth

              for i=1:4
                I1=Nrb+i;
                Bt=Bt+Cof(i)*(B(Ith+1,I1)-B(Ith-1,I1));
                Brt=Brt+Cof(i+4)*(B(Ith+1,I1)-B(Ith-1,I1));
                Btt=Btt+Cof(i)*(B(Ith+1,I1)+B(Ith-1,I1)-2*B(Ith,I1));
              end
              Bint1(Ith) = Bint;
              Bt=Bt/(2.0*Dths);
              Bt1(Ith) = Bt;
              Brt=Brt/(Dr*2.0*Dths);
              Btt=Btt/(Dths*Dths);
              Brt1(Ith) = Brt;
              Btt1(Ith) = Btt;
            end

            Pts=Psq-Y(2)*Y(2); %Pts = Ptheta^2
            if (Pts <= 0)
                fprintf('Error Pr=P & Pause');
                pause;
            end
            Pt=sqrt(Pts);
            Ptsc(Istp+1,j)=Pts;

            % R and x equations.
              Xc0=Stp*Y(1)/Pt; % Xc0 = r/Pth*dtheta. Y(1)=r
              Ck(1)=Xc0*Y(2);  % dr = rPr/Pth*dtheta, Y(2)=Pr
              Ck(2)=Stp*(Pt-Y(1)*Bint); % dPr = (Pth - rB)*dtheta
              Xc1=Stp*Y(2)/Pt;  % Xc1 = Pr/Pth*dtheta
              Xc2=Psq*Xc0/Pts;  % Xc2 = P*P*r/(Pth*Pth*Pth)*dtheta
              Xc3=Stp*(Brint+Bint); % Xc3 = (rdB/dr+B)*dtheta
              Ck(3)=Xc1*Y(3)+Xc2*Y(4); %dx=(Pr/Pth*x+rP^2/Pth^3*Px)*dtheta
              Ck(4)=-Xc1*Y(4)-Xc3*Y(3); %dPx=(-Pr/Pth*x-
(B+r*dBr/dr)*x)*dtheta
              Ck(5)=Xc1*Y(5)+Xc2*Y(6); %Y(3)=1,Y(4)=0,Y(5)=0,Y(6)=1
              Ck(6)=-Xc1*Y(6)-Xc3*Y(5); %Y(3)=1,Y(4)=0,Y(5)=0,Y(6)=1
              Neq=6;
            % Phase, Z, and Rav equations (Last time only)
              if (Ntry == -2 )
                Neq=13;
                Xc4=Stp*Brint-Xc1*Bt; % rBr/dr*dtheta-
Pr/Pth*dtheta*(dB/dth)
                Ck(7)=Xc0*Gam-Stp; % Phase Error in closed orbit
                Ck(8)=Xc0*Y(9); % dz = r/Pth*dtheta*Pz
                Ck(9)=Xc4*Y(8); % dPz = (rBr/dr*dtheta-Pr/Pth*dB/dth)*z
```

```
                Ck(10)=Xc0*Y(11); %Y(8)=1,Y(9)=0,Y(10)=0,Y(11)=1
                Ck(11)=Xc4*Y(10); %Y(8)=1,Y(9)=0,Y(10)=0,Y(11)=1
                Ck(12)=Stp*Y(1); % r*dtheta = (R average)*theta
                Ck(13)=Xc0*Gam; % r/Pth*dtheta = tau in closed orbit
                % fprintf('Ptheta=%f, Pr=%f, Ck(10)=%f, Y(1)=%f,
Y(11)=%f\n',Pt,Y(2),Ck(10),Y(1),Y(11));
              end

              if (Ntry==-2 && order==3)
                 Neq=4;
                 for k=1:Npar
                     Y1f=YY(1,k);
                     DRf = 0.5; %%%%%%%%%%%%%%%
                     Frbf=(Y1f-R0)/DRf-1; % Y1=r R0=1 Dr=0.5
                   %  fprintf('Frbf=%f\n',Frbf);
                     if (Frbf < 0)
                     fprintf('Error Off field k-th particle\n',k);
                     YY(1,k)=0;YY(2,k)=0;YY(3,k)=0;YY(4,k)=0;
                     continue;
                     end
                     Nrbf=fix(Frbf); % Frb = 2*(r-1)-1 = 2r-3 ??
                     Frbf=Frbf-Nrbf;

                     if ( Nrbf+4-Nr > 0)
                     fprintf('Error Off field k-th particle\n',k);
                     YY(1,k)=0;YY(2,k)=0;YY(3,k)=0;YY(4,k)=0;
                     continue;
                     end
                     %
                     % R-interpolation for B-field
                     Frbf2=Frbf*Frbf;
                     Frbf3=Frbf*Frbf2;
                     Coff(1)=Frbf2-0.5*(Frbf3+Frbf);
                     Coff(2)=1.5*Frbf3-2.5*Frbf2+1.0;
                     Coff(3)=2.0*Frbf2-1.5*Frbf3+0.5*Frbf;
                     Coff(4)=0.5*(Frbf3-Frbf2);
                     Coff(5)=2.0*Frbf-1.5*Frbf2-0.5;
                     Coff(6)=4.5*Frbf2-5.0*Frbf;
                     Coff(7)=4.0*Frbf-4.5*Frbf2+0.5;
                     Coff(8)=1.5*Frbf2-Frbf;
                     Coff(9)=2.0-3.0*Frbf;
                     Coff(10)=9.0*Frbf-5.0;
                     Coff(11)=4.0-9.0*Frbf;
                     Coff(12)=3.0*Frbf-1;

                     Bintf=0.0; % B
                     Brf=0.0; % dB/dr
                     Brrf=0.0; % d^2B/dr^2

                     for i=1:4
                        Btif=B(Ith,Nrbf+i);
                        Bintf=Bintf+Coff(i)*Btif;
                        Brf=Brf+Coff(i+4)*Btif;
                        Brrf=Brrf+Coff(i+8)*Btif;
                     end
                     Brf = Brf/DRf;
                     Brf1(Ith) = Brf;
                     Brrf=Brrf/(DRf*DRf);
```

```
                    Brrf1(Ith) = Brrf;
                    % Theta derivative (Only when z equations used)
                    Btf=0.0; % dB/dth
                    Bttf=0.0; % d^2B/dth^2
                    Brtf=0.0; % d^2B/drdth

                    for i=1:4
                      I1=Nrbf+i;
                      Btf=Btf+Coff(i)*(B(Ith+1,I1)-B(Ith-1,I1));
                      Brtf=Brtf+Coff(i+4)*(B(Ith+1,I1)-B(Ith-1,I1));
                      Bttf=Bttf+Coff(i)*(B(Ith+1,I1)+B(Ith-1,I1)-
2*B(Ith,I1));
                    end
                    Bintf1(Ith) = Bintf;
                    Btf=Btf/(2.0*Dths);
                    Btf1(Ith) = Btf;
                    Brtf=Brtf/(DRf*2.0*Dths);
                    Bttf=Bttf/(Dths*Dths);
                    Brtf1(Ith) = Brtf;
                    Bttf1(Ith) = Bttf;
                   Ptsf=Psq-YY(2,k)^2-YY(4,k)^2;
                   if(Ptsf<0) fprintf('Ptsf=%f, Ptsf must be
positive\n',Ptsf); close; end
                    Ptf=sqrt(Ptsf);
                 % fprintf('YY(2,k)=%f, YY(4,k)=%f,
Ptf=%f\n',YY(2,k),YY(4,k),Ptf);
                   Ckk(1,k)=YY(1,k)*YY(2,k)/Ptf*Stp; % dr =
rPr/Pth*dtheta, YY(1,k)=r, YY(2.k)=Pr
                   Ckk(2,k)=(Ptf-YY(1,k)*(Bintf-0.5*YY(3,k)^2*(Brrf
+1/YY(1,k)*Brf+1/(YY(1,k)^2)*Bttf))+YY(3,k)*YY(4,k)*Btf/Ptf)*Stp;
                   % dPr/dtheta = Pt-r(B-
1/2*z^2*gradient^2(B))+z*Pz*(dB/dth)/Pt
                   Ckk(3,k)=YY(1,k)*YY(4,k)/Ptf*Stp; % dz =
rPz/Pth*dtheta  YY(1,k)=r, YY(4,k)=Pz
                   Ckk(4,k)=YY(3,k)*(YY(1,k)*Brf-YY(2,k)/Ptf*Btf)*Stp;
                   % dPz/dtheta = z*(r*(dB/dr)-Pr/Pt*(dB/dth));
                 end
               end
               for i=1:Neq
                 Rrk=Ark(j)*(Ck(i)-Brk(j)*Q(i));
                 Y(i)=Y(i)+Rrk;
                 Q(i)=Q(i)+3.0*Rrk+Crk(j)*Ck(i); % Runge-Kutta Gill method
               end
               if (Ntry==-2 && order==3)
                 for i=1:Neq
                   for k=1:Npar
                     Rrkk=Ark(j)*(Ckk(i,k)-Brk(j)*QQ(i,k));
                     YY(i,k)=YY(i,k)+Rrkk;
                     QQ(i,k)=QQ(i,k)+3.0*Rrkk+Crk(j)*Ckk(i,k); % Runge-
Kutta Gill method
                   end
                 end
               end
          end %for j=1:4(RK) end

          if (Ntry==-2 && order==2)
              X11=Y(3); X12=Y(5);
              X21=Y(4); X22=Y(6);
```

```
            Z11=Y(8); Z12=Y(10);
            Z21=Y(9); Z22=Y(11);
            X(1,1)=X11; X(1,2)=X12; X(2,1)=X21; X(2,2)=X22;
            Z(1,1)=Z11; Z(1,2)=Z12; Z(2,1)=Z21; Z(2,2)=Z22;
            a12=Psq/Pt^3; % alpha_12 = P^2/Pth^3
            a13=1.5*Y(1)*Y(2)*Psq/(Pt^5); %alpha_13=3/2*r*Pr*P^2/Pth^5
            a16=0.5*Y(1)*Y(2)/Pt^3; %alpha_16=1/2*r*Pr/Pth^3
            f(1,1)=a12*X11*X21+a13*X21^2;
            f(1,2)=a12*(X11*X22+X12*X21)+2*a13*X21*X22;
            f(1,3)=a12*X12*X22+a13*X22^2;
            f(1,4)=a16*X21^2;
            f(1,5)=2.0*a16*Z21*Z22;
            f(1,6)=a16*Z22^2;

            a21=-0.5*(2.0*Br+Y(1)*Brr); %-1/2*(2*dB/dr+rd^2B/dr^2)
            a23=-0.5*Psq/Pt^3; %-1/2*P^2/Pth^3
            a24=0.5*Y(1)*(Brr
+1/Y(1)*Br+1/Y(1)^2*Btt); %1/2*r*(gradient^2(B))
            a25=Bt/Pt; %(dB/dth)/Pth
            a26=-0.5/Pt; %-1/2*1/Pth
            f(2,1)=a21*X11^2+a23*X21^2;
            f(2,2)=2.0*a21*X11*X12+2.0*a23*X21*X22;
            f(2,3)=a21*X12^2+a23*X22^2;
            f(2,4)=a24*Z11^2+a25*Z11*Z21+a26*Z21^2;
            f(2,5)=2.0*a24*Z11*Z12+a25*(Z11*Z22+Z12*Z21)+2.0*a26*Z21*Z22;
            f(2,6)=a24*Z12^2+a25*Z12*Z22+a26*Z22^2;

            b12=1/Pt; % beta_12=1/Pth;
            b14=Y(1)*Y(2)/Pt^3; % beta_14=r*Pr/Pth^3
            b21=Br+Y(1)*Brr-Y(2)/Pt*Brt; %beta21 = dB/dr+r*d^2B/dr^2-
Pr/Pt*d^2B/drdt
            b23=-Psq/Pt^3*Bt; %beta_23=-P^2/Pt^3*dB/dth
            g(1,1)=b12*X11*Z21+b14*X21*Z21;
            g(1,2)=b12*X11*Z22+b14*X21*Z22;
            g(1,3)=b12*X12*Z21+b14*X22*Z21;
            g(1,4)=b12*X12*Z22+b14*X22*Z22;
            g(2,1)=b21*X11*Z11+b23*X21*Z11;
            g(2,2)=b21*X11*Z12+b23*X21*Z12;
            g(2,3)=b21*X12*Z11+b23*X22*Z11;
            g(2,4)=b21*X12*Z12+b23*X22*Z12;

            gam2=Gam*Y(2)/Pt^3; %gamma_2=gamma*Pr/Pth^3;

gam3=0.5*Gam*Y(1)/Pt^3*(1+3.0*Y(2)^2/Pt^2); %gamma_3=1/2*gamma*r/Pth^3*(1+3
*Pr^2/Pth^2)
            gam6=0.5*Gam*Y(1)/Pt^3; %gamma_6=1/2*gamma*r/Pth^3;
            h(1)=gam2*X11*X21+gam3*X21^2;
            h(2)=gam2*(X11*X22+X12*X21)+2.0*gam3*X21*X22;
            h(3)=gam2*X21*X22+gam3*X22^2;
            h(4)=gam6*Z21^2;
            h(5)=2.0*gam6*Z21*Z22;
            h(6)=gam6*Z22^2;

   %          fprintf('INT_GZ(1,2)=%f\n',INT_GZ(1,1));
            for i=1:6
              INT_GX(:,i)=INT_GX(:,i)+inv(X)*f(:,i)*Stp;
            end
      %       fprintf('before iStp=%f, INTGZ(1,1)=%f\n',Istp,INT_GZ(1,1));
```

```
            for i=1:4
               INT_GZ(:,i)=INT_GZ(:,i)+inv(Z)*g(:,i)*Stp;
            end
      %      fprintf('after iStp=%f, INTGZ(1,1)=%f\n',Istp,INT_GZ(1,1));
            for i=1:6
               GX(:,i)=X*INT_GX(:,i);
            end
            for i=1:4
               GZ(:,i)=Z*INT_GZ(:,i);
            end
            if(cal_cen==1)
              Xi = zeros(1,2); Zi = zeros(1,2);
              Xf = zeros(2,Npar); Zf = zeros(2,Npar);
             for i = 1:Npar
              Xi(1)=xi(i); Xi(2)=xpi(i); Zi(1)=zi(i); Zi(2)=zpi(i);
              Vx(1)=xi(i)^2; Vx(2)=xi(i)*xpi(i); Vx(3)=xpi(i)^2;
              Vx(4)=zi(i)^2; Vx(5)=zi(i)*zpi(i); Vx(6)=zpi(i)^2;
             Vz(1)=xi(i)*zi(i); Vz(2)=xi(i)*zpi(i);
             Vz(3)=xpi(i)*zi(i); Vz(4)=xpi(i)*zpi(i);

              Xft(:,i,Istp+1)=X*Xi'+ GX*Vx';
              Zft(:,i,Istp+1)=Z*Zi'+ GZ*Vz';
             end
          end
        end
        if (Ntry == -2)
           Delt=Y(13)-Time1;
           Bavet=Bavet+Bint*Delt;
        end

        if (Istp ~= 1)
           if (Y(5)*Yp5 < 0.0)
              Nxcros = Nxcros + 1;
           end
           if (Y(10)*Yp10 < 0.0)
              Nzcros = Nzcros + 1;
           end
        end

      end %Istp end


      if (Ntry == -2)
          Ntry = -1;
      end

      if (Ntry ~= -1)
         Ep1 = Y(1)-Rg;
         Ep2 = Y(2)-Prg;


% Modified by S. Shin (2016. 02. 17)
% Check conv relation later.
          Den = Y(3)+Y(6)-2;
          Rg = Rg + (Ep1*(Y(6)-1.0)-Ep2*Y(5))/Den;
          Prg = Prg + ((Y(3)-1.0)*Ep2-Y(4)*Ep1)/Den;
          conv = abs(Ep1)+abs(Ep2)-1.d-1*Rg;
```

```
        if (conv < 0)
           Ntry = -2;
        end
       end

      if (KK == Iter)
        fprintf('Error not converge');
        stop;
      end
    end %Iter end

    Rst(Nstp+1)=Y(1);
    Prst(Nstp+1)=Y(2);
    fprintf('Rst=%f,Prst=%f\n',Rst(Nstp+1),Prst(Nstp+1));

% Particles tracking
%
    if (order==1)
      Xf = zeros(2,Npar);
      Zf = zeros(2,Npar);

      for i=1:Npar
        Xf(1,i) = Y(3)*xi(i) + Y(5)*xpi(i);
        Xf(2,i) = Y(4)*xi(i) + Y(6)*xpi(i);

        Zf(1,i) = Y(8)*zi(i) + Y(10)*zpi(i);
        Zf(2,i) = Y(9)*zi(i) + Y(11)*zpi(i);
      end

      if (ieo==plot_eo)

          figure;
          xlim([-0.15 0.15]);
          ylim([-0.06 0.06]);
          hold on
          plot(Xf(1,:),Xf(2,:),'bo');
          title('X vs Xp at Center E.O. 1st order','FontSize',16)
          xlabel('X','FontSize',16)
          ylabel('Xp','FontSize',16)

          figure;
          xlim([-0.15 0.15]);
          ylim([-0.06 0.06]);
          hold on
          plot(Zf(1,:),Zf(2,:),'bo');
          title('Z vs Zp at Center E.O. 1st order','FontSize',16)
          xlabel('Z','FontSize',16)
          ylabel('Zp','FontSize',16)

          figure;
          xlim([-0.6, 0.6]);
          ylim([-0.6, 0.6]);
          hold on
          plot(Xf(1,:),Zf(1,:),'bo');
          title('X vs Z at Center E.O. 1st order','FontSize',16)
          xlabel('X','FontSize',16)
```

```
        ylabel('Z','FontSize',16)

       %% Calculate RMS emittance for first order
        Ax=sum(Xf(1,:).^2)/5000;  % <x^2>
        Apx=sum(Xf(2,:).^2)/5000; % <x'^2>

        Xxf=Xf(1,:)';
        Axpx2=Xf(2,:)*Xxf/5000;   % <xx'>
        Axpx2=Axpx2^2;           % <xx'>^2
        Erms=sqrt(Ax*Apx-Axpx2)   % RMS emittance root(<x^2><x'^2> -
<xx'>^2)


        Az=sum(Zf(1,:).^2)/5000;  % <z^2>
        Apz=sum(Zf(2,:).^2)/5000; % <z'^2>

        Zzf=Zf(1,:)';
        Azpz2=Zf(2,:)*Zzf/5000;   % <zz'>
        Azpz2=Azpz2^2;           % <zz'>^2
        Ermsz=sqrt(Az*Apz-Azpz2)  % RMS emittance root(<z^2><z'^2> -
<zz'>^2)
     end
   end

   if (order==2)
      Xi = zeros(1,2); Zi = zeros(1,2);
      Xf = zeros(2,Npar); Zf = zeros(2,Npar);
      for i = 1:Npar
        Xi(1)=xi(i); Xi(2)=xpi(i); Zi(1)=zi(i); Zi(2)=zpi(i);
        Vx(1)=xi(i)^2; Vx(2)=xi(i)*xpi(i); Vx(3)=xpi(i)^2;
        Vx(4)=zi(i)^2; Vx(5)=zi(i)*zpi(i); Vx(6)=zpi(i)^2;
        Vz(1)=xi(i)*zi(i); Vz(2)=xi(i)*zpi(i);
        Vz(3)=xpi(i)*zi(i); Vz(4)=xpi(i)*zpi(i);

        Xf(:,i) = X*Xi' + GX*Vx';
        Zf(:,i) = Z*Zi' + GZ*Vz';

        Xf1(:,i) = GX*Vx';
        Zf1(:,i) = GZ*Vz';

      end

      if (ieo==plot_eo)
         fprintf('second z=%f\n',Zf(1,1));
        figure;
        xlim([-0.15 0.15]);
        ylim([-0.06 0.06]);
        hold on
        plot(Xf(1,:),Xf(2,:),'bo');
        title('X vs Xp at Center E.O. 2nd order','FontSize',16)
        xlabel('X','FontSize',16)
        ylabel('Xp','FontSize',16)

        figure;
        xlim([-0.15 0.15]);
        ylim([-0.06 0.06]);
        hold on
```

```
        plot(Zf(1,:),Zf(2,:),'bo');
        title('Z vs Zp at Center E.O. 2nd order','FontSize',16)
        xlabel('Z','FontSize',16)
        ylabel('Zp','FontSize',16)

        figure;
        xlim([-0.15 0.15]);
        ylim([-0.06 0.06]);
        hold on
        plot(Xf(1,:),Zf(1,:),'bo');
        title('X vs Z at Center E.O. 2nd order','FontSize',16)
        xlabel('X','FontSize',16)
        ylabel('Z','FontSize',16)

      %% Calculate RMS emittance for second order
        Ax=sum(Xf(1,:).^2)/5000;  % <x^2>
        Apx=sum(Xf(2,:).^2)/5000; % <x'^2>

        Xxf=Xf(1,:)';
        Axpx2=Xf(2,:)*Xxf/5000;   % <xx'>
        Axpx2=Axpx2^2;           % <xx'>^2
        Erms=sqrt(Ax*Apx-Axpx2)   % RMS emittance root(<x^2><x'^2> -
<xx'>^2)


        Az=sum(Zf(1,:).^2)/5000;  % <z^2>
        Apz=sum(Zf(2,:).^2)/5000; % <z'^2>

        Zzf=Zf(1,:)';
        Azpz2=Zf(2,:)*Zzf/5000;   % <zz'>
        Azpz2=Azpz2^2;           % <zz'>^2
        Ermsz=sqrt(Az*Apz-Azpz2)  % RMS emittance root(<z^2><z'^2> -
<zz'>^2)
      end
   end

   if (order==3)
      Xf = zeros(2,Npar);
      Zf = zeros(2,Npar);
        for i=1:Npar
          Xf(1,i)=YY(1,i)-Y(1);
          Xf(2,i)=YY(2,i)-Y(2);
          Zf(1,i)=YY(3,i);
          Zf(2,i)=YY(4,i);
        end
        if (ieo == plot_eo)
          figure;
          xlim([-0.15 0.15]);
          ylim([-0.06 0.06]);
          hold on
          plot(Xf(1,:),Xf(2,:),'bo');
          title('X vs Xp at Center E.O. full','FontSize',16)
          xlabel('X','FontSize',16)
          ylabel('Xp','FontSize',16)

          figure;
          xlim([-0.15 0.15]);
```

```
            ylim([-0.06 0.06]);
            hold on
            plot(Zf(1,:),Zf(2,:),'bo');
            title('Z vs Zp at Center E.O. full','FontSize',16)
            xlabel('Z','FontSize',16)
            ylabel('Zp','FontSize',16)

            figure;
            xlim([-0.6, 0.6]);
            ylim([-0.6, 0.6]);
            hold on
            plot(Xf(1,:),Zf(1,:),'bo');
            title('X vs Z at Center E.O. full','FontSize',16)
            xlabel('X','FontSize',16)
            ylabel('Z','FontSize',16)

         %% Calculate RMS emittance for full equation
           Ax=sum(Xf(1,:).^2)/5000;  % <x^2>
           Apx=sum(Xf(2,:).^2)/5000; % <x'^2>

           Xxf=Xf(1,:)';
           Axpx2=Xf(2,:)*Xxf/5000;   % <xx'>
           Axpx2=Axpx2^2;           % <xx'>^2
           Erms=sqrt(Ax*Apx-Axpx2)   % RMS emittance root(<x^2><x'^2> -
<xx'>^2)


           Az=sum(Zf(1,:).^2)/5000;  % <z^2>
           Apz=sum(Zf(2,:).^2)/5000; % <z'^2>

           Zzf=Zf(1,:)';
           Azpz2=Zf(2,:)*Zzf/5000;   % <zz'>
           Azpz2=Azpz2^2;           % <zz'>^2
           Ermsz=sqrt(Az*Apz-Azpz2)  % RMS emittance root(<z^2><z'^2> -
<zz'>^2)
         end
    end

% Calculatioin r and z tunes.

    for i = 1:2
       I1 = 5*(i-1);
       Cn=0.5*(Y(3+I1)+Y(6+I1));
       Sn=1.0 - Cn*Cn;

       if (Sn < 0)
% Unstable case that is abs(Cn) > 1.
          Rznt = -log(abs(Cn)+sqrt(-Sn));
       else

          Sn=sqrt(Sn);

          if (Y(5+I1) < 0)
% x2 z2 < 0, what's mean?
             Sn = -Sn;
          end
```

```
        Rznt = atan2(Sn,Cn);

        if (Rznt < 0)
            Rznt = Rznt + 6.283185;
        end
      end
      Rznu(i)=Rznt/(Nsint*Sect);

   end

   if ((E-P(1,2))/P(2,2) >=5 )
      Foe=Foe+3141.5927*P(2,2)*(Phsp+Phs);
   end
   Stt(ieo)=ieo;
   Ec(ieo)=E;
   Rgc(ieo)=Rg;
   Rznu1(ieo)=Rznu(1);
   Rznu2(ieo)=Rznu(2);
   Bavec(ieo)=Bave;

   % Next energy and new R,Pr guess.
   E=E+P(2,2);
   ieo=ieo+1;
   if (ieo - Neo <= 0)
      Ep1 = 0.5*Rg*P(2,2)/(E-P(2,2));
      Ep2 = 0.0;
   else
       Ep1 = Rg-Rp;
       Ep2 = Prg - Prp;
   end
   Rp = Rg;
   Rg = Rg +Ep1; %initial position of next orbit
   Prp = Prg;
   Prg = Prg +Ep2; %intial momentum of next orbit

   time = datestr(now);
   hour2 = str2double(time(13:14));
   min2 = str2double(time(16:17));
   sec2 = str2double(time(19:20));

   fprintf('# Computing time = %d hour %d minute %d sec at %d-th
turn\n',hour2-hour1,min2-min1,sec2-sec1,ieo-1);
end  %ieo end
```

7.5 Acceleration code description

The variables are the same as those in sec. 7.4. To run this code, refer to variables of sec. 7.4.

```
% Start program
% define varibles
    .
    .
    . %refer variables of sec7.4.

for Ieo = 1:Neo

    Iter = 20;
    KK = 0; % Number of iteration steps.
    Psq = E/E0;
    Gam = 1.D+0 + Psq;
    Psq = Psq*(2.D+0 + Psq)*Acon^2;

       % Set initial values of integration variables.
       Y(1) = Rg;    % Initial position.
       Y(2) = Prg;   % Initail momentum.
       Y(3) = 1.D+0; % x1
       Y(4) = 0.D+0; % px1
       Y(5) = 0.D+0; % x2
       Y(6) = 1.D+0; % px2
       Y(7) = 0.D+0;
       Y(8) = 1.D+0; % z1
       Y(9) = 0.D+0; % pz1
       Y(10)= 0.D+0; % z2
       Y(11)= 1.D+0; % Pz2
       Y(12)= 0.D+0;
       Y(14)=0;
       for I =1:14
          Q(I)=0.0;
       end

       % Loop for finding orbit at each theta.
       for Istp = 1:Nstp %180¹ø µ¹¸²(2µµ¾¿ 360µµ)
          Istp=Istp-1;
       % Begin integration step.
       % Ntry = 0 at the begining.
          Rst(Istp+1)=Y(1);  % Position
          Prst(Istp+1)=Y(2); % Momentum
          Y3(Ieo,Istp+1)=Y(3);
          Y4(Ieo,Istp+1)=Y(4);
          Y5(Ieo,Istp+1)=Y(5);
          Y6(Ieo,Istp+1)=Y(6);
          Y8(Ieo,Istp+1)=Y(8);
          Y9(Ieo,Istp+1)=Y(9);
          Y10(Ieo,Istp+1)=Y(10);
          Y11(Ieo,Istp+1)=Y(11);
          Yp5=Y(5);
          Yp10=Y(10);

          % Plot Orbit for each energy.

            Cox(Ieo,Istp+1)=Rst(Istp+1);
```

```
            Cop(Ieo,Istp+1)=Prst(Istp+1);
            coxx(Ieo,Istp+1)=2.54*Cox(Ieo,Istp+1)*cos((Istp+1)*2*pi/180);
            coyy(Ieo,Istp+1)=2.54*Cox(Ieo,Istp+1)*sin((Istp+1)*2*pi/180);
            corr(Ieo,Istp+1)=sqrt(coxx(Ieo,Istp+1)*coxx(Ieo,Istp+1) +
coyy(Ieo,Istp+1)*coyy(Ieo,Istp+1));
            Time1= Y(13);


          %
          % Runge-Kutta iteration
          %
          for j = 1:4
              Ith=fix(j/2);

              % Ith is from 2 to 362.
              % ..... 357, 357, 358, 358, 359, 359, ...
              Ith=2*Istp+Ith+2;

              while (Ith >= Nth+2)
                  Ith=Ith-Nth;
              end
              % Y(1) is initial value at starting theta
              Y1=Y(1);

              Frb=(Y1-R0)/Dr-1;

              if (Frb < 0) % The condition of Y1 < 1.
                  fprintf('Error Off field');
                  stop;
              end

              Nrb=fix(Frb);
              Frb=Frb-Nrb;

              if ( Nrb+4-Nr > 0) % Maybe out of field range.
                  fprintf('Error Off field');
                  stop;
              end
              %
              % R-interpolation for B-field
              % Here used fields in equations are Bint, Brint and Btint.
              %
              Frb2=Frb*Frb;
              Frb3=Frb*Frb2;

              Cof(1)=Frb2-0.5*(Frb3+Frb);
              Cof(2)=1.5*Frb3-2.5*Frb2+1.0;
              Cof(3)=2.0*Frb2-1.5*Frb3+0.5*Frb;
              Cof(4)=0.5*(Frb3-Frb2);
              Cof(5)=2.0*Frb-1.5*Frb2-0.5;
              Cof(6)=4.5*Frb2-5.0*Frb;
              Cof(7)=4.0*Frb-4.5*Frb2+0.5;
              Cof(8)=1.5*Frb2-Frb;
              Bint=0.0;
              Brint=0.0;

              for i=1:4
```

```
            Bti=B(Ith,Nrb+i);
            Bint=Bint+Cof(i)*Bti;
            Brint=Brint+Cof(i+4)*Bti;
        end

        Bint;
        Brint=Brint*Y1/Dr;

        % Theta derivative (Only when z equations used)

       Btint=0.0;
       Ith1=Ith-1;
       Ith2=Ith+1;
       for i=1:4
          I1=Nrb+i;
          Btint=Btint+Cof(i)*(B(Ith2,I1)-B(Ith1,I1));
       end
       Btint=Btint/(2.0*Dths);
       Btest(Ieo,Istp+1)=Btint; % Test

        Pts=Psq-Y(2)*Y(2);

        if (Pts <= 0)
      %     fprintf('Error Pr=P');
      %     stop;
        end
        Pt=sqrt(Pts);
        Ptsc(Istp+1,j)=Pts;

        % R and x equations.
        % Phase, Z, and Rav equations (Last time only)

        Neq=13;
        Xc0=Stp*Y(1)/Pt;

        Ck(1)=Xc0*Y(2); % dr=(rPr/Pt)*dth. dr=Ck(1), Y(1) is r, dt is
Stp and Y(2) is Pr.
        Ck(2)=Stp*(Pt-Y(1)*Bint); % dPr=(Pt-q'rB)*dth. Where is q'?
        Xc1=Stp*Y(2)/Pt;
        Xc2=Psq*Xc0/Pts;
        Xc3=Stp*(Brint+Bint);
        Ck(3)=Xc1*Y(3)+Xc2*Y(4); % dx1 =(Pr*x1/Pt+r*P^2*Px1/Pt^3)*dth
        Ck(4)=-Xc1*Y(4)-Xc3*Y(3); % dPx1 = ...
        Ck(5)=Xc1*Y(5)+Xc2*Y(6); % dx2 =(Pr*x2/Pt+r*P^2*Px2/Pt^3)*dth
        Ck(6)=-Xc1*Y(6)-Xc3*Y(5); % dPx2 = ...

          Xc4=Stp*Brint-Xc1*Btint;
          Ck(7)=Xc0*Gam-Stp; % dph = (r/Pt)*Gam*dth*m'-dth. Here m'
is 1.Check it.
          Ck(8)=Xc0*Y(9); % dz1=(rPz1/Pt)*dth
          Ck(9)=Xc4*Y(8); % dPz1= ...
          Ck(10)=Xc0*Y(11); % dz1= ...
          Ck(11)=Xc4*Y(10); % dPz2= ...
          Ck(12)=Stp*Y(1); % Average radus ??
          Ck(13)=Xc0*Gam; % Average field ??

        %acceleration
```

```
                Gapd=10;
                Gapd2=80;

                if (Istp == Gapd) | (Istp == Gapd+90);
                    econst=Chg*Xc0*Gam*Acon^2/E0;
                    Etheta=0.045/(2*pi*corr(Ieo,Istp)/180); %1cm gap, 45kV
                    Ck(14)=Chg*Xc0*(Pt*Etheta);
                    Neq=14;
                end

                if (Istp == Gapd2) | (Istp == Gapd2+90);
                    econst=Chg*Xc0*Gam*Acon^2/E0;
                    Etheta=0.045/(2*pi*corr(Ieo,Istp)/180); %1cm gap, 45kV
                    Ck(14)=Chg*Xc0*(Pt*Etheta);
                    Neq=14;
                end

                for i=1:2
                   Rrk=Ark(j)*(Ck(i)-Brk(j)*Q(i));
                   Y(i)=Y(i)+Rrk;
                   Q(i)=Q(i)+3.0*Rrk+Crk(j)*Ck(i);
                end

                for i=3:Neq
                    Rrk=Ark(j)*(Ck(i)-Brk(j)*Q(i));
                    Y(i)=Y(i)+Rrk;
                    Q(i)=Q(i)+3.0*Rrk+Crk(j)*Ck(i);
                end

            end

            if (Istp == Gapd) | (Istp == Gapd+90) | (Istp == Gapd2) | (Istp
== Gapd2+90);
                E=E+Y(14);
                Psq = E/E0;
                Gam = 1.D+0 + Psq;
                Psq = Psq*(2.D+0 + Psq)*Acon^2;
            end
            % End of Runge-Kutta iteration
%
% Additional module for calculating phase space motion.
             xx(Ieo,Istp+1)=Rst(Istp+1);
             ppxx(Ieo,Istp+1)=Prst(Istp+1);
              Energy(Ieo,Istp+1)=E;
              Y3(Ieo,Istp+2)=Y(3);
              Y4(Ieo,Istp+2)=Y(4);
              Y5(Ieo,Istp+2)=Y(5);
              Y6(Ieo,Istp+2)=Y(6);
              Y8(Ieo,Istp+2)=Y(8);
              Y9(Ieo,Istp+2)=Y(9);
              Y10(Ieo,Istp+2)=Y(10);
              Y11(Ieo,Istp+2)=Y(11);

              % Phase space for multi particle: Opt=1
              % Phase space for single particle: Opt=2

              Opt = 1;
```

```
            if (Opt ==1)

              phx(Ieo,Istp+1,:)=Y3(Istp+1)*xi(Ieo,:)+Y5(Istp+1)*pxi(Ieo,:);

phpx(Ieo,Istp+1,:)=Y4(Istp+1)*xi(Ieo,:)+Y6(Istp+1)*pxi(Ieo,:);

phz(Ieo,Istp+1,:)=Y8(Istp+1)*zi(Ieo,:)+Y10(Istp+1)*pzi(Ieo,:);

phpz(Ieo,Istp+1,:)=Y9(Istp+1)*zi(Ieo,:)+Y11(Istp+1)*pzi(Ieo,:);

            %   k=k+1;
             %  evpx(k,:)=phx(Ieo,Istp+1,:);
              % evppx(k,:)=phpx(Ieo,Istp+1,:);
               %evpz(k,:)=phz(Ieo,Istp+1,:);
               %evppz(k,:)=phpz(Ieo,Istp+1,:);


              if Istp ==179
                xi(Ieo+1,:)=phx(Ieo,Istp+1,:); % initial condition
                pxi(Ieo+1,:)=phpx(Ieo,Istp+1,:); % initial condition
                zi(Ieo+1,:)=phz(Ieo,Istp+1,:); % initial condition
                pzi(Ieo+1,:)=phpz(Ieo,Istp+1,:); % initial condition
              end

            else
                xi(1)=0.1;
                pxi(1)=0.0;
                zi(1)=0.1;
                pzi(1)=0.0;
                phx(Ieo,Istp+1)=Y3(Istp+1)*xi(Ieo)+Y5(Istp+1)*pxi(Ieo);
                phpx(Ieo,Istp+1)=Y4(Istp+1)*xi(Ieo)+Y6(Istp+1)*pxi(Ieo);
                phz(Ieo,Istp+1)=Y8(Istp+1)*zi(Ieo)+Y10(Istp+1)*pzi(Ieo);
                phpz(Ieo,Istp+1)=Y9(Istp+1)*zi(Ieo)+Y11(Istp+1)*pzi(Ieo);

                k=k+1;
                evpx(k)=phx(Ieo,Istp+1);
                evppx(k)=phpx(Ieo,Istp+1);
                evpz(k)=phz(Ieo,Istp+1);
                evppz(k)=phpz(Ieo,Istp+1);

              if Istp ==179
                xi(Ieo+1)=phx(Ieo,Istp+1); % initial condition
                pxi(Ieo+1)=phpx(Ieo,Istp+1); % initial condition
                zi(Ieo+1)=phz(Ieo,Istp+1); % initial condition
                pzi(Ieo+1)=phpz(Ieo,Istp+1); % initial condition
              end

            end
%         End of additional module.
 %
 % End loop for finding orbit at each theta.
   Ed(Ieo)=Energy(Ieo,180)-Energy(Ieo,1);
   Rst(Nstp+1)=Y(1);
   Prst(Nstp+1)=Y(2);
   %X1
```

```
    %Fixed Chul-Un find last degree.

              Cox(Ieo,Nstp+1)=Rst(Nstp+1);
              Cop(Ieo,Nstp+1)=Prst(Nstp+1);
              coxx(Ieo,Nstp+1)=2.54*Cox(Ieo,Nstp+1)*cos((Nstp+1)*2*pi/180);
              coyy(Ieo,Nstp+1)=2.54*Cox(Ieo,Nstp+1)*sin((Nstp+1)*2*pi/180);
              corr(Ieo,Nstp+1)=sqrt(coxx(Ieo,Nstp+1)*coxx(Ieo,Nstp+1) +
coyy(Ieo,Nstp+1)*coyy(Ieo,Nstp+1));
    %
    % Compute focusing frequencies and print normal output line.
    %
    Phsp=Phs;
    Phs = Y(7)/(Nsint*Sect); % Phase error
    Rav = Y(12)/(Nsint*Sect); % Average radius
    Bave = Bavet*Bcon/Y(13); % Average field

    for i = 1:2
       I1 = 5*(i-1);
       Cn=0.5*(Y(3+I1)+Y(6+I1));
       Sn=1.0 - Cn*Cn;

       if (Sn < 0)
          Rznt = -log(abs(Cn)+sqrt(-Sn));
       else
          Sn=sqrt(Sn);
          if (Y(5+I1) < 0)
             Sn = -Sn;
          end

          Rznt = atan2(Sn,Cn);

          if (Rznt < 0)
             Rznt = Rznt + 6.283185;
          end
       end

       if (i ==1 && fix(Nsint*Sect)==6)
          Rznu(i)=Rznt/(Nsint*Sect)+1;
       else
          Rznu(i)=Rznt/(Nsint*Sect);
       end

    end

    if ((E-P(1,2))/P(2,2) >=5 )
       Foe=Foe+3141.5927*P(2,2)*(Phsp+Phs);
    end
%    fprintf('   %f   %f   %f   %f   %f   %f   %f   %f   %f   %f
\n',E,2.54*Rav,2.54*Rg,Prg,Phs,Rznu(1),Nxcros,Rznu(2),Nzcros,Foe,Bave);


%
% Initial Twiss parameter
%

    Cntx=0.5*(Y(3)+Y(6));
    Sntx=1.0 - Cntx*Cntx;
```

```
    Sntx=sqrt(Sntx);
    betax0(Ieo)=Y(5)/Sntx;
    gammax0(Ieo)=-Y(4)/Sntx;
    alphax0(Ieo)=0.5*(Y(3)-Y(6))/Sntx;

    Cnty=0.5*(Y(8)+Y(11));
    Snty=1.0 - Cnty*Cnty;
    Snty=sqrt(Snty);
    betay0(Ieo)=Y(10)/Snty;
    gammay0(Ieo)=-Y(9)/Snty;
    alphay0(Ieo)=0.5*(Y(8)-Y(11))/Snty;
%
% Calculating beta function
%
   for i=1:Nstp
      betax(Ieo,i)=Y3(i)*Y3(i)*betax0(Ieo)-
2*Y5(i)*Y3(i)*alphax0(Ieo)+Y5(i)*Y5(i)*gammax0(Ieo);
      betay(Ieo,i)=Y8(i)*Y8(i)*betay0(Ieo)-
2*Y10(i)*Y8(i)*alphay0(Ieo)+Y10(i)*Y10(i)*gammay0(Ieo);
   end

    Ravc(Ieo)=Rav;
    Ec(Ieo)=E;
    Rgc(Ieo)=Rg;
    Phsc(Ieo)=Phs;
    Rznu1(Ieo)=Rznu(1);
    Rznu2(Ieo)=Rznu(2);
    Bavec(Ieo)=Bave;
    Foec(Ieo)=Foe;

    % Go to next orbit.
    Ieo=Ieo+1;  % Next Orbit
    Rg = Rst(Nstp+1); % Initial position of nth orbit
    Prg = Prst(Nstp+1); % Initial Momentum of nth orbit
end
     .
     .
     .
     .
     .
     ...
```

8. References

[1] Abderrahim, H. Ait, et al. “MYRRHA: A Multipurpose Accelerator Driven System for Research & Development,” Nucl. Instrum. and Meth. A 463, p.487 (2001).

[2] G. Dutto, et al. “Recent Performance of the TRIUMF Cyclotron and Status of the Facility,” No. TRI-PP-95-79, SCAN-9511213 (1995).

[3] M. Seidel, et al. “Production of a 1.3 MW Proton Beam at PSI,” *proceedings of IPAC10*, 1309 (2010).

[4] M. K. Craddock, “AG Focusing in the Thomas Cyclotron of 1938,” *Proceedings of PAC09* (2009)

[5] Kim, Y. S., An, D. H., Chai, J. S., Chang, H. S., Hong, B. H., Hong, S. S., … and Kim, J. H. “New Design of the KIRAMS-13 Cyclotron for Regional Cyclotron Center,” *Proceedings of APAC* (2004).

[6] S. Bernal, “A Practical Introduction to Beam Physics Particle and Particle Accelerators”, A Morgan & Claypool Publication, second ed.

[7] M. Conte, W.W. Mackay, “An Introduction to the Physics of Particle Accelerators,” World Scientific, second ed.

[8] Y. Jongen, S. Zaremba, “Cyclotron Magnet Calculations,” *Proceedings of the CERN-1996-02*, p.139 (1996).

[9] F. Chautrad, “Beam dynamics for cyclotrons,” CERN Accelerator School (2005).

[10] H.A. Hagedoorn, N.F. Verster, “Orbit in An AVF Cyclotron,” Nucl. Instrum. and Meth 18-19, 201 (1962).

[11] M.M. Gordon, “Computation of Closed Orbits and Basic Focusing Properties for Sector-Focused cyclotrons and The Design of “CYCLOPS”,” Part. Accel. 16, 39 (1984).

[12] Choi, C. U., Chung, M., Hahn, G.R., Lee, J., Lee, T. Y., and Shin, S. “Simulation Code Development for High-Power Cyclotron,” 21st Int. Conf. on Cyclotrons and Their Applications (Cyclotrons’ 16), Zurich, Switzerland, September 11-16, 2016, p.68-70 (2017).

[13] B.F. Milton, “Phase Selection in the K500 Cyclotron and The Development of a Non-Linear Transfer Matrix Program,” Ph.D. thesis, Michigan University (1986).

[14] Hyun Wook Kim, Joonsun Kang, Bong Hwan Hong, In Su Jung, “Design study of the KIRAMS-430 superconducting cyclotron magnet,” Nucl. Instrum. and Meth. Vol. 823, p. 26-31, ISSN 0168-9002, https://doi.org/10.1016/j.nima.2016.03.113 (2016).

[15] B.F. Milton, “CYCLONE VERS 8.4,” TRI-DN-99-4, TRIUMF report (1999).

[16] An. D. H., Chai, J. S., Chang, H. S., Hong, B. H., Jung, I. S., Kang, J., … and Yang, T. K. “Beam Trajectory Simulations for KIRAMS-13 Cyclotron,” *Proceedings of APAC* (2004).

Acknowledgement

Foremost, I would first like to express my gratitude to my supervisor, Prof. M. Chung for the helping me finish the master's course for two years in UNIST. His guidance helped me in all the time of research and writing of this thesis. I could not have imagined having a better advisor and mentor for my master's study.

Besides my advisor, I'd like to also thank to Dr. S. Shin and Dr. B.H. Oh in Pohang accelerator Laboratory, and Dr. G. Hahn in Korea Institute of Radiological & Medical Science. I was able to acquire a lot of knowledge about accelerator physics because you kindly directed me. And I was so happy to be able to work with you who kind people.

I'm also grateful to our lab members who have lived and enjoyed for two years in IBAL.

At last, I'd like to thank my parents for always trusting me and supporting me, I am grateful to all my friends who had listened to me when I was in trouble.

Through my two years of laboratory experience in master's degree, I think that physical thinking has developed as a physicist. Although I do not go to doctorate, I will study physics forever. Thank you.